

# IMG2IMU: Applying Knowledge from Large-Scale Images to IMU Applications via Contrastive Learning

Hyungjun Yoon  
hyungjun.yoon@kaist.ac.kr  
School of Electrical Engineering,  
KAIST  
Republic of Korea

Hyeongheon Cha  
chahh98@kaist.ac.kr  
School of Electrical Engineering,  
KAIST  
Republic of Korea

Canh Hoang Nguyen  
canhhoang30011999@gmail.com  
School of Computing, KAIST  
Republic of Korea

Taesik Gong  
taesik.gong@kaist.ac.kr  
School of Computing, KAIST  
Republic of Korea

Sung-Ju Lee  
profsj@kaist.ac.kr  
School of Electrical Engineering,  
KAIST  
Republic of Korea

## Abstract

Recent advances in machine learning showed that pre-training representations acquired via self-supervised learning could achieve high accuracy on tasks with small training data. Unlike in vision and natural language processing domains, such pre-training for IMU-based applications is challenging, as there are only a few publicly available datasets with sufficient size and diversity to learn generalizable representations. To overcome this problem, we propose *IMG2IMU*, a novel approach that adapts pre-train representation from large-scale images to diverse few-shot IMU sensing tasks. We convert the sensor data into visually interpretable spectrograms for the model to utilize the knowledge gained from vision. Further, we apply contrastive learning on an augmentation set we designed to learn representations that are tailored to interpreting sensor data. Our extensive evaluations on five different IMU sensing tasks show that *IMG2IMU* consistently outperforms the baselines, illustrating that vision knowledge can be incorporated into a few-shot learning environment for IMU sensing tasks.

## 1 Introduction

Advancements in ubiquitous computing utilizing deep learning have enabled mobile data collected in everyday life to be used in numerous applications. Particularly, motion sensing with internal measurement units (IMU), such as accelerometers and gyroscopes, has emerged as a major strand of mobile sensing due to its vast array of applications: activity recognition [8, 40, 41, 45, 51, 66, 72], driving context inference [6, 25, 31], eating detection [22, 63, 77], machine fault detection [13, 37], and medical applications [16, 48]. Sensing applications typically employ deep learning models that are trained through supervision from task-specific datasets. For such a setting, the performance of the model depends

heavily on both the quality and quantity of the training data. However, in real deployments, acquiring a large amount of data for sensing applications is challenging due to the data collection cost and privacy issues, and thus model training is often performed with insufficient data.

Recent research has delved into effectively training deep learning models with limited training data, such as pre-training models to teach knowledge for general tasks (i.e. representation learning), then fine-tuning them with data from the actual downstream tasks [5]. An effective method for pre-training the representation model is self-supervised learning, which uses large amounts of unlabeled data to learn the characteristics of the data with the predefined pretext task. This strategy shows remarkable performance in numerous domains with large public datasets. As an example in natural language processing, BERT [19], a pre-trained model without labels from a large-scale data corpus such as Wikipedia, is used as a foundation model for various NLP tasks. In computer vision, the models that are fine-tuned from the model pre-trained through self-supervised learning on ImageNet [60], which consists of 1,000 classes and 1.2M instances, are achieving state-of-the-art performance on several classification tasks [10].

Applying self-supervised learning to IMU sensing applications has been proposed [26, 27, 61, 76, 78]. This line of research showed that pre-training with unlabeled IMU sensor data improves downstream sensing performance. However, prior works focus mostly on Human Activity Recognition (HAR) tasks, and research on sensing tasks other than HAR is still underexplored. The main reason for this absence could be attributed to the lack of variety and quantity from publicly available IMU sensing data. Contrary to computer vision [49, 60] and natural languages [58, 81] where massive-scale public datasets have been established, data types and

scales of mobile sensing are limited by the public directories, making it hard for developers to use representation learning. Public datasets for sensors are centered on HAR [8, 45, 66, 72], covering a few activities such as walking and sitting, which are typically collected from controlled in-lab environments. A publically available large scale [20] data is collected only from the user’s smartwatch at a single sampling rate, and hence it lacks diversity in the type, position, and signal processing method of the sensing device. A pre-trained model trained on limited dataset results in scalability issues, making it difficult to be fine-tuned on downstream tasks showing different characteristics. Therefore, self-supervised learning from public sensor data would show poor generalizability due to the (i) limited tasks centered on HAR and (ii) the lack of diversity in the sensing device, position, and user pool.

In light of the limitation of public IMU sensor data, our idea is motivated by a question: *Should the representation for sensors necessarily be learned only from the sensor data?* The data collected by sensors can be represented in the form of images, such as spectrograms [1, 34, 39]. When the data is transformed into 2D, the interpretation of informative features in visual form can be supported by common knowledge of interpreting images as pattern recognition, object detection, and color recognition. This implies IMU sensing applications can benefit from the learned representations from diverse and large-scale datasets in vision domain.

In order to facilitate training with limited data on diverse IMU sensing deployments, we present *IMG2IMU* that uses and adapts the representation trained from public image data, such as ImageNet [60], as the base model. In *IMG2IMU*, sensor data is represented as spectrograms, which is one of the most widely used 2d-transformation methods. Three sensor axes are mapped to the RGB channels of the converted spectrogram image. In subsequent steps, the learned representation from public images is utilized to fine-tune the downstream task with the converted sensor data.

In *IMG2IMU*, a novel image-based pre-training is utilized to learn a representation suitable for IMU sensing. A domain gap exists between the knowledge required to interpret sensor data and the knowledge required to interpret images derived from public datasets. As an example, rotating an image 90 degrees to the left seems to depict the same image from a different viewpoint in vision; however, the frequency and time axes are reversed in the spectrogram image. To prevent such cases and minimize the domain gap, we suggest a contrastive learning approach using sensing-specialized image augmentations as the pre-training method. As part of contrastive learning, four augmentations are used to generate positive samples, TranslateX, PermutateX, Hue, and Jitter, which help the model learn the sensory properties that are needed to interpret sensor data.

We demonstrate that the proposed pre-training with sensor-specialized augmentations improves the robustness of the model against the sensory variabilities. Further, we evaluated *IMG2IMU* on a variety of IMU sensing tasks and found it outperformed the baselines consistently in the scenario of learning with limited data is available for training. When it was evaluated on the IMU sensing tasks other than HAR, *IMG2IMU* showed up to 6.02% higher performance in mean F1-score compared with the existing self-supervised learning method designed for HAR.

We summarize the main contributions of the paper as follows:

- We propose *IMG2IMU* that utilizes a model pre-trained from public image dataset through self-supervised learning to improve IMU sensing applications.
- Based on the domain knowledge of sensors, we present a set of image augmentations that enables contrastive learning on images to learn knowledge that is useful for sensing tasks.
- We provide analysis of how each augmentation affects the self-supervised model to be robust against sensory data properties.
- We demonstrate through experiments that *IMG2IMU* improves the performance of diverse sensing tasks where only limited data is available for training.

## 2 Related Work

### 2.1 Self-Supervised Learning for Sensing

Recent works in self-supervised learning have brought significant advances in the sensing field, primarily for human activity recognition (HAR). Through the use of pretext tasks that can be designed without labels, self-supervised learning enables the model to learn the general properties of the data. It has been suggested that specially designed pretext tasks can be used to capture the properties from unlabeled sensor data [27, 42, 61, 76].

Multi-task learning [7] is a popular approach for pretext tasks in self-supervised learning. Prior works [61, 78] applied self-supervised learning using multi-task Transformation Prediction Network (TPN) for HAR. They defined an augmentation set that can be applied to sensor data. Using TPN, the original data is augmented with one random augmentation and the network is trained to predict the type of augmentation applied.

Contrastive learning is another effective method for a pretext task. Contrastive learning generates positive samples from data by applying augmentations that preserve the key invariant features and trains the model to learn that the embedding from the original data point is similar to the positive samples and dissimilar to the other data

points. MoCo [11, 12, 28] and SimCLR [9, 10] are representative contrastive learning frameworks and they have been redesigned for HAR tasks (MoCoHAR [73], SimCLR for HAR [67], and CSSHAR [42]). Haresamudram et al. [27] adopted Contrastive Predictive Coding (CPC) [52] to pre-train models with unlabeled data. In CPC, an encoder is trained to predict the next sequence chunk based on the previous sequence chunks. Masked region prediction [74], more commonly known as BERT’s learning strategy [19], also adopts self-supervised learning. Several research [26, 76] have leveraged masked region prediction by adapting the learning strategy to sensor data.

While those self-supervised learning studies have demonstrated their effectiveness for HAR tasks and the possibility of learning with a small amount of labeled data, the approach to pre-train a model on sensor data has a fundamental limitation; *a large amount of unlabeled sensor data is necessary to pre-train a model*. However, publicly available IMU sensing data are few and far between, compared with traditional ML domains such as computer vision. Moreover, most accessible IMU sensor data are limited to human activity recognition (HAR) tasks. Consequently, most previous research on self-supervised learning on sensing applications used HAR datasets [8, 20, 45, 66, 72].

However, applications utilizing IMU sensors involve different target tasks [6, 25, 37], different users and behaviors [40, 41, 51], different sensors, and different data collection methods (i.e., sensor position and sampling frequency) [16, 63]. As publically available datasets lack such diversity, we hypothesize that the pre-trained model for sensing has poor generalizability to diverse sensing tasks. Our evaluation confirms that the model trained on publicly available HAR data suffers from performance degradation when applied to other sensing tasks via transfer learning.

Our approach circumvents this challenge by interpreting IMU sensor data as images. Once 2d-transformed data is used as input for sensing tasks, the tasks can be viewed as mere image classification tasks. As representation learning from large-scale image datasets has shown to be generalizable in a variety of image classification tasks [29, 32], we hypothesize it to work with the converted sensor data.

## 2.2 Use of Cross-Modal Data for Sensing

To improve self-supervised learning for IMU sensing, learning with data from different modalities has been proposed. ColloSSL [38] and COCOA [17] use sensor data in cross-modal, measured from different modalities, for contrastive learning. Vision2Sensor [57] proposed an approach of vision-to-sensor label transmission to learn sensor data measured concurrently through the labels generated by vision-based activity recognition. Nevertheless, previous works use a learning method that interprets other modalities of sensing as

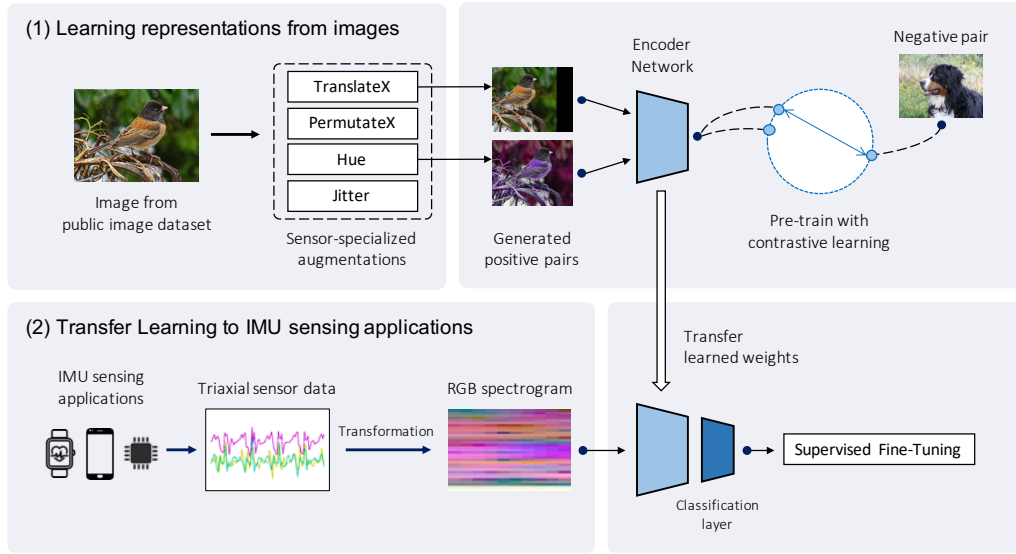
another view of the same context, and thus synchronization of data is essential. This requirement is problematic as it is difficult to collect multi-modality data in the same context as in public IMU sensor data. Another line of research is virtual IMU sensor data generation. IMUTube [46, 47] presented a framework for generating virtual IMU sensor data from publicly available videos. By mapping virtual IMU sensors to the estimated subject position in videos, IMUTube generates virtual sensor data. However, the components of IMUTube are mostly designed only for HAR, and the virtual data generation is strongly affected by screen occlusions and video quality. Thus, current virtual IMU data generation cannot generate sufficient sensor data for wider range of tasks. In *IMG2IMU* on the other hand, it only requires the existing public image dataset for pre-training, and pre-trained and fine-tuned datasets are separated and learned independently, thus synchronization between data is not needed.

## 2.3 Using Pre-Trained Models from Images

Models pre-trained on large-scale image datasets, such as ImageNet [60] or COCO [49], have proven their effectiveness in transfer learning [29, 32]. In computer vision, these pre-trained models show high performance in a diverse range of tasks, including image classification [21], object detection [24], and semantic segmentation [14]. With the use of self-supervised learning as a pre-training strategy, transfer learning from pre-trained models becomes more powerful. It was demonstrated that pre-training using self-supervised learning, which enables the use of unlabeled data, shows superior performance in transfer learning compared with pre-training using supervised learning [10].

There has been extensive use of image-based pre-trained models across many domains that differ from the images contained in the public dataset. Azizi et al. [3] utilized the model that is pre-trained on ImageNet for medical image classification, using the images for dermatology skin classification [50] and multi-label chest X-ray classification [36]. The pre-trained model from ImageNet is even used for sound classification tasks [64], which was enabled by converting the sound data into mel-spectrogram that can be interpreted as images. Prior studies have indicated that using the pre-trained model as the foundation could result in improvements for tasks from different domains.

Inspired by these findings, *IMG2IMU* uses the ImageNet pre-trained model for IMU sensing applications. To apply knowledge from images, we use the triaxial property of the IMU sensor data to map each axis of data to the RGB channel of the image. *IMG2IMU* differs from prior studies in that it uses a novel pre-training method to learn a representation that involves knowledge specific to sensor data interpretation, rather than simply applying pre-trained knowledge



**Figure 1: Overview of *IMG2IMU*.** (1) First, contrastive learning is performed using images based on the selected augmentations. (2) Then trained weights are fine-tuned with spectrograms converted from triaxial IMU sensor data.

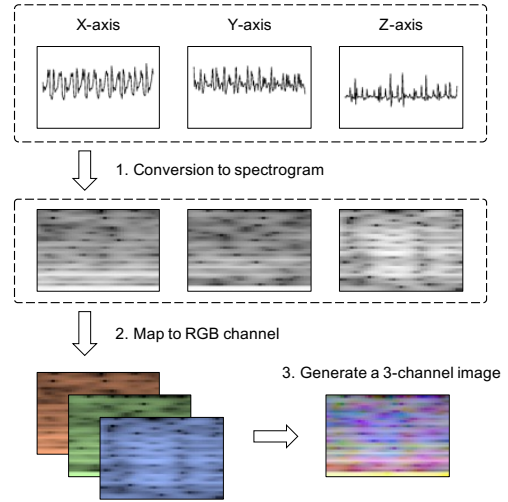
from images. Section 3.3 elaborates our design of pre-training for IMU sensing applications.

### 3 *IMG2IMU*

To enhance the performance of IMU sensing tasks where large-scale training data are often difficult to obtain, we propose to utilize public image datasets to pre-train a base model. Our approach is grounded in the intuition that sensor data can be presented in a form of images and interpreted with general knowledge of image recognition, such as pattern recognition. Figure 1 shows the overview of our proposed *IMG2IMU*. *IMG2IMU* consists of four main components: (i) converting sensor data into 2D representations, (ii) pre-training the model using public image data through self-supervised contrastive learning, (iii) adapting the augmentations during contrastive learning so that the model learns useful sensor-specific knowledge, and (iv) transferring the learned knowledge from the public images to downstream sensing tasks. We detail each component in the following.

#### 3.1 Converting IMU Sensing Data to Images

Various methods have been used to transform sensor data into 2D [1, 15, 23, 34, 39, 59, 80]. Sensing data is often represented in the form of images to improve interpretability or employ 2D CNN structures, which is known to provide high accuracy in computer vision [1, 59]. Previous research [34] has shown that 2D representations of sensors contain enough features to be used as the input for sensing applications.



**Figure 2: Generation of 3-channel 2D image from IMU sensor data.**

We aim to utilize a model that has been pre-trained from a large-scale public image dataset to support the interpretation of the 2D representations of the sensor data. To match the input shape for the pre-trained model, we convert sensor data originally represented as 1D time-series into 2D spectrograms [59], which is a widely known transformation method for sensor data. As a form of 2D representation of time-series data, the spectrogram displays the intensity of a frequency feature along the time axis. Frequency features are known

to play an important role in sensing tasks, and spectrograms display them directly as images. We thus expect that the ability to interpret visual features from images can also be applied to spectrograms.

To generate data from IMU sensors, commonly triaxial sensors are used to indicate motion along all three axes, x, y, and z. The input data for the model should include all of the spectrogram data for each channel as IMU sensing generally requires the information of all three channels. Considering that image data commonly consist of three color channels (i.e., the RGB channel), we map the x, y, and z channels from triaxial data to the R, G, and B channels when generating images. The method is based on a conversion method found to be successful in previous studies [1, 59]. Figure 2 shows the overall process to generate a 3-channel 2D image from IMU sensor data. Spectrograms are created for each channel and converted into corresponding color channels to create images. Afterward, the generated image is resized and normalized to fit the model’s input shape.

## 3.2 Contrastive Self-Supervised Learning

In the field of computer vision, contrastive learning [9, 28] has emerged as one of the most promising methods for self-supervised learning. Contrastive learning trains the model with unlabeled data by learning a strategy for maximizing agreement between positive pairs that closely match and discriminating other negative pairs from them. Recent success on contrastive learning has proven that the learned representation from contrastive learning works as a strong semi-supervised learner for downstream tasks [10]. As the main pre-training strategy of *IMG2IMU*, we adopt contrastive learning because of not only their remarkable performance but also the nature of utilizing positive and negative pairs. In contrastive learning, the model learns the mutual information between the positive pairs to differentiate them from the negative pairs. This means we can control what the model learns by handling the positive pairs. Focusing on this point, we design a contrastive learning method to learn the representation specialized to interpret sensor data. We discuss the detailed method in Section 3.3.

To implement contrastive learning in *IMG2IMU*, we adopt a popular self-supervised learning baseline in MoCo [28]. We chose MoCo as our main framework, as when compared with other contrastive learning methods such as SimCLR [9], MoCo uses a much smaller batch size while still achieving comparable performances. This allows model developers to use the framework in more diverse, resource-constrained environments, resulting in greater scalability.

MoCo creates a positive pair from its training data that provides a different *view* by applying a random data augmentation. The model is then trained to distinguish the positive pair from the negative samples that are generated from other

data points. InfoNCE loss [52] is used to train the encoder that generates the query  $q$  from the training data, positive sample key  $k_+$  from the positive pair, and other negative samples’ keys  $k_i (i = 0, 1, 2, \dots, K)$ . The training objective is to distinguish the positive sample key ( $k_+$ ) from the other negative keys ( $k_i$ ). We can calculate the InfoNCE loss used for contrastive learning in MoCo as follows:

$$L_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^K \exp(q \cdot k_i / \tau)}, \quad (1)$$

where  $\tau$  indicates the temperature parameter. MoCo maintains a large set of negative keys by constructing a dictionary that stores data of multiple encoded keys. A moving average is used to update the key encoder, which allows the dictionary to remain dynamic. After contrastive learning is performed on the training image data, the parameters of the query encoder network are used as pre-trained weights for the downstream sensing task.

## 3.3 Augmentation Selection

*3.3.1 Selecting Sensor-Specialized Augmentations* In contrastive learning, data augmentation plays a key role; it produces positive samples from unlabeled data via the deformation of images. Data augmentation preserves the *invariant*, i.e., the key property of the data, and generates a different *view* of the same data. As an example, images are often rotated, flipped, and scaled to change their viewpoint while maintaining color and relative shapes. In addition, due to the fact that images can appear differently depending on the lighting or camera settings, augmentations such as changing brightness and contrast are used while preserving features related to the object’s shape. Using augmentations in contrastive learning, the model learns what mutual information to use to cognize the original and augmented data as the same.

The types of augmentations should be carefully selected based on what knowledge the model aims to acquire in contrastive learning. It has been shown that different augmentations can vary in their usefulness to different downstream tasks [70]. For instance, when training a model for a downstream task that classifies arrow signs indicating different directions, augmentations such as flip or rotate would confuse the model in learning directional properties, which might be the most important feature for classification.

Our downstream task takes spectrogram images derived from IMU sensor data as the input. Compared with the images used for pre-training, spectrogram images show different properties. To handle the domain gap between them, the type of augmentations for generating positive views should be carefully selected. For example, spectrograms have *directional properties along the axes*, thus common vision augmentations such as flipping vertically or horizontally would harm

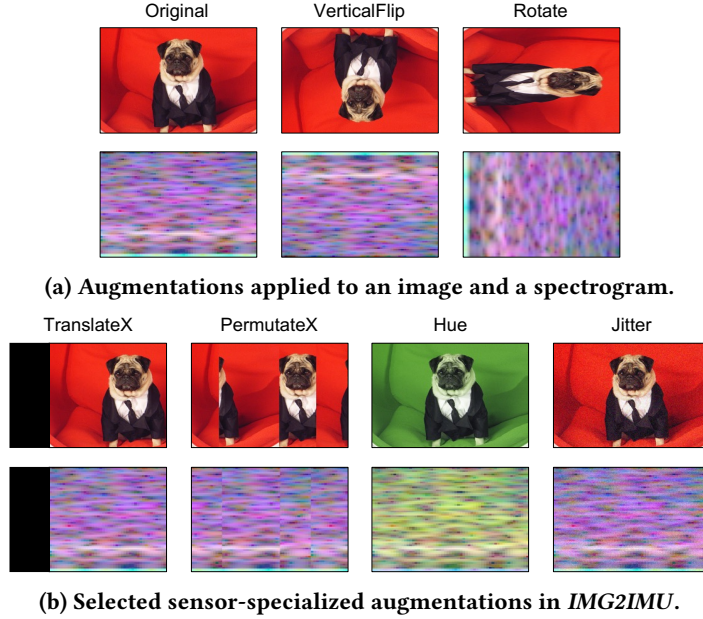


Figure 3: Visualization of the effect of augmentations when applied to an image and a spectrogram.

the downstream performance as they generate reversed view of the recorded sensor data. Further, augmentations such as rotation would distort the nature of the spectrogram representation as *each axis has fixed values of time and frequency*.

Figure 3a illustrates some widely-used image augmentations applied to a public dataset image and a spectrogram image. As opposed to preserving the label information of the public image, the augmentations convert the spectrogram image to data with different frequency features. To prevent such problems caused by the domain gap between the public image data and the spectrogram, we refer to the invariants that should be considered in the sensing tasks and define an augmentation set to assist the model in learning useful knowledge for downstream sensory tasks. We list the augmentations that we found highly correlated with sensory tasks and thus useful for the positive sample generation.

- **TranslateX:** TranslateX randomly shifts image data on the x-axis. For downstream sensing tasks, sensor data are usually segmented into fixed-size windows. During this stage, the window can be started at any point in the data as long as it is from the same label. In other words, as the key features of data are within the window, the classification remains the same regardless of whether a window is shifted left or right over the time axis. Based on this property, we assume that the augmentation that translates the image on the x-axis will have a beneficial effect on sensory tasks as the x-axis represents time in the spectrogram.
- **PermutateX:** PermutateX splits the original data over the x-axis into multiple chunks and randomly perturbs the split chunks. As an augmentation method for sensory data, permutation is known to preserve the local temporal features while distorting the global structure of the data to produce a different view. Existing work [71] has shown that permutation is one of the most beneficial augmentation methods for sensors. PermutateX directly replicates the permutation augmentation by fitting the temporal perturbation into the x-axis.
- **Hue:** Hue alters the color tone of the entire image while preserving the overall brightness and contrast. The values between RGB channels are often interchanged with this augmentation. In IMU sensing, x, y, and z channels are interchangeable based on the rotation of the sensor. To reflect the diversity, rotation is commonly used as an augmentation method for triaxial sensors [71]. The x, y, and z channels of the sensor data are mapped to the RGB channel of an image using our approach. By applying the Hue augmentation, we can replicate the effect of interchangeability between the three channels in the triaxial sensor data.
- **Jitter:** Jitter adjusts the color by adding random noise for each of the pixels in the image. We injected uniform noise centered on zero to implement this augmentation. Jitter mimics the augmentation method of adding random noise to waveform sensory data. Sensors can be affected by random noise, which can affect the spectrogram by making some regions brighter or darker depending on

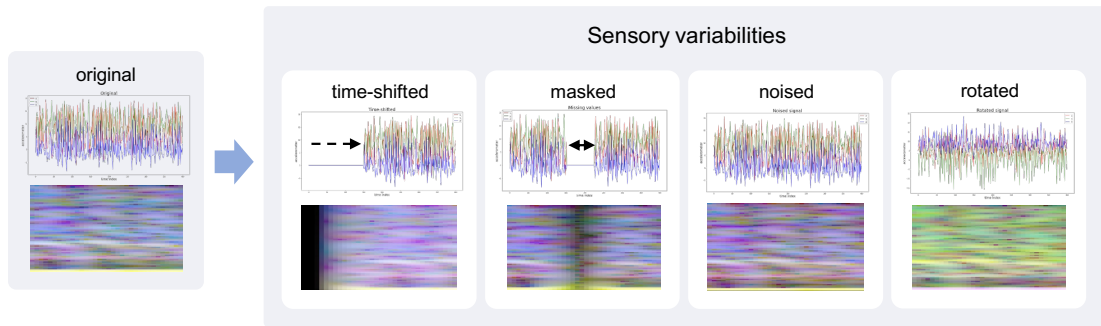


Figure 4: The synthetic data created from the WISDM dataset. From the original data, we generated four types of augmented data to replicate possible deformations of sensor data: time-shifted, masked, noised, and rotated. The images below indicate the converted spectrograms from the sensor data.

Table 1: Evaluation of the effect of each augmentation on the robustness against the deformations. Tx, Px, H, and J denotes TranslateX, PermuteX, Hue, and Jitter respectively. The pre-trained model using all augmentations (TxPxHJ) and models except for each augmentation are compared. We report the drop of F1-score in each deformation compared to the clean data. The largest drop shown in F1-score ( $\pm 1\%$ ) is bold.

	original	time-shifted		masked		noised		rotated	
	F1	F1	drop	F1	drop	F1	drop	F1	drop
TxPxHJ (Default)	0.7544	0.5450	27.75%	0.5344	29.16%	0.5801	23.10%	0.6953	7.83%
PxHJ (w/o Tx)	0.6861	0.4338	<b>36.78%</b>	0.4678	31.83%	0.6266	8.68%	0.6838	0.34%
TxHJ (w/o Px)	0.6874	0.4354	<b>36.66%</b>	0.3866	<b>43.76%</b>	0.5328	22.48%	0.6606	3.90%
TxPxH (w/o J)	0.7488	0.5401	27.87%	0.5015	33.02%	0.5622	<b>24.91%</b>	0.6949	7.19%
TxPxJ (w/o H)	0.7040	0.5585	20.68%	0.5475	22.24%	0.6224	11.59%	0.5389	<b>23.45%</b>

the hardware or environment. We adopt Jitter to make the model robust to the random noise that could be included in sensory data from uncontrolled settings.

Figure 3b visualizes the selected image augmentations applied to both an image from a public dataset and a spectrogram.

**3.3.2 Effect of Augmentations on Downstream Tasks** Our choice of augmentations for contrastive learning is based on four sensor-specific invariants: TranslateX correlates with features *robust to time-shifts in sensor data*, PermuteX correlates with *preserving local temporal features*, Hue correlates with *robustness to interchangeability of data between axes*, and Jitter correlates with *robustness to noise*.

We conducted an experiment to understand and visualize the aforementioned augmentation-wise correlations for downstream sensory tasks. For this experiment, we pre-trained a baseline model on ImageNet dataset with contrastive learning using the four augmentations (TranslateX, PermuteX, Hue, and Jitter). In addition, we pre-trained four extra models, where each model trained using three augmentations except one augmentation per model:

(TranslateX, PermuteX, Hue), (TranslateX, PermuteX, Jitter), (TranslateX, Hue, Jitter), and (PermuteX, Hue, Jitter).

For the sensory dataset, we used WISDM [45] which is one of the most popular human activity recognition datasets. The models are fine-tuned and evaluated on the WISDM dataset. Additionally, for the evaluation datasets, we used extra four synthetic data generated from WISDM. They were generated to simulate intra-class variabilities that can occur in sensor data. The purpose of using synthetic data in this experiment is to figure out the augmentations that affect the robustness against different types of deformations in sensor data. If pre-training with specific augmentation shows a clear performance increase in certain distortion, it can be said that the augmentation positively affects the robustness against the distortion. As the deformations, (i) time-shifted data was generated by shifting the sensor to left or right, (ii) masked data was generated to simulate the internal disconnections within a window of sensor data, (iii) noised data was made by adding random noise to the original data, and (iv) rotated data was generated by applying a linear transformation to interchange values across axes. Figure 4 shows

the synthetic sensor data generated from WISDM and the converted spectrograms from the data.

In Table 1, mean F1-scores for each pre-trained model is shown for each synthetic dataset. The first column presents the performance when the trained models are evaluated on the WISDM dataset. We can observe that the pre-trained models show varying performance depending on the augmentations used in contrastive learning. Overall, the pre-trained model with the full augmentation set performs the best, and as we eliminate each augmentation, performance drops to different degrees. To determine whether the pre-trained models are robust to sensor data variabilities, we compare their performances on synthetic datasets. Considering that each augmentation enhances the performance of the original dataset, we examine the performance drop rate when each model is tested on different synthetic datasets.

First, in the time-shifted dataset, the models trained without TranslateX and PermuteX show the largest performance drop rates. It matches with our hypothesis that TranslateX affects the robustness towards time-shift in the sensor data, and thus removing TranslateX from the augmentation set brings a significant drop when the data is shifted. PermuteX preserves the local temporal features of the sensors, and as the robustness to time-shift requires those features, removing PermuteX degrades the performance. The effect of removing PermuteX is prominent when data is masked. We masked the data to distort the global features of the sensor data. In such cases, the classification should be done by utilizing only the remaining local features. With the masked data, the augmentation set without PermuteX yielded a significant performance drop, verifying that PermuteX enhances the use of local features with the pre-trained model.

We noticed that compared to the other augmentations, removing Jitter from the full augmentation set results in a small performance drop. However, with noisy dataset, the pre-trained model without Jitter shows the largest drop. This result indicates Jitter enhances the robustness towards noise. Finally, with the rotated data, eliminating Hue from the augmentation set clearly harms the robustness towards rotation, with extremely lower performance than other models.

Based on the experimental results, we verified our hypothesis on the correlation between the augmentations and the invariants of sensor data. We learned that TranslateX, PermuteX, Hue, and Jitter can be used as a valid set of augmentations. Our subsequent evaluations on IMU sensing tasks are based on the selected set of augmentations. Our experiments in Section 4.2.3 demonstrate that our selected augmentation set outperforms other augmentation sets that do not consider sensor properties.

### 3.4 Fine-Tuning to Sensing Tasks

We follow the traditional fine-tuning setup in self-supervised learning studies [9, 28]; the model trained through self-supervised learning on the public image dataset is fine-tuned on a small subset of data from each downstream sensory task. As illustrated in Figure 2, the data from downstream tasks, which are from IMU sensing applications, are represented as spectrograms. *IMG2IMU* evaluates the model based on two different approaches for fine-tuning. For the first approach, we freeze the backbone networks and attach a trainable fully connected layer as the linear classifier at the end of the backbone network. In this approach, the backbone model works as a fixed feature extractor, allowing us to evaluate the quality of the pre-trained representation transmitted by the frozen backbone layers [4, 44, 52, 79].

Another approach to fine-tune the model is to update its entire parameters including the backbone networks and the classifier. This setting is used to examine the end-to-end transferability of the pre-trained model to downstream tasks. To train our network in the end-to-end transfer learning setting, we employ a two-step training procedure: freezing the backbone networks and training only the classifier layer for the first few epochs, followed by training the entire network in the subsequent epochs [69]. In this manner, our model converges to optimal weights in a short period of time.

## 4 Evaluation

### 4.1 Experimental Settings

**4.1.1 Datasets** *IMG2IMU* uses images to pre-train representations for downstream sensing tasks. As the pre-training dataset, we employ ImageNet (ILSVRC2012) [60], a widely known image dataset. In ImageNet, a total of 1.28 million images are used as the training dataset, involving 1,000 classes. For our training, the labels are not used as we use self-supervised learning without label information.

We evaluate the effectiveness of the pre-trained model from images with five different sensing datasets. The IMU sensing datasets are all for classification tasks and contain triaxial accelerometer data. We consider a variety of tasks to investigate the generalizability of *IMG2IMU*. Table 2 provides an overview of the datasets. For each dataset, 10 *shots* (i.e., 10 samples for each class) is used to fine-tune the model and the remaining data is used for evaluation.

**WISDM [45].** This dataset covers the basic human activity recognition task. Six different activities of sitting, standing, walking, jogging, walking downstairs, and walking upstairs, were performed by 36 participants. During the experiment, participants carried smartphones in their pockets and accelerometer data was collected from the smartphone at a sampling rate of 20 Hz. We used a window size of 600 in our experiments.



**Table 2: Overview of datasets used in our evaluation.**

Task	Sensing device	Sampling rate	Subject	Sample	Classes
WISDM	Smartphone internal IMU	20 Hz	36	9,303	6
Eating Detection	ADXL313	400 Hz	24	50,000	2
Road Anomaly Detection	Smartphone internal IMU	50 Hz	13	1,156	2
Fetal Movement Detection	MPU 9250	280 Hz	13	22,300	2
Goat Activity Recognition	ProMove-mini	100 Hz	5	30,748	5

**Road Anomaly Detection [25].** This dataset is used to classify anomalies in roads, including potholes, speed bumps, metallic bumps, and other instances. Triaxial acceleration data is measured using different types of smartphones located in vehicles. A total of 13 types of vehicles were used and the smartphones were positioned in five different locations inside the car. The sampling rate of the data is 50Hz. We only use two classes of data, pothole and speed bump, as other classes do not have enough samples. The experiment is conducted with a window size of 300.

**Goat Activity Recognition [40].** This dataset describes basic activity recognition for goats in farms. Data was collected by six triaxial accelerometers (ProMove-mini [68]) attached to the collar-shaped device worn by five goats. The dataset includes five activities; stationary, walking, eating, running, and trotting. Data was sampled at a rate of 100 Hz and we used a window size of 900.

**Eating Detection [63].** This data contains sensor data measured for sensing food intakes. To measure sensor data, ADXL313 [18] triaxial accelerometer was attached to an eye-glass. Twenty four participants wore the glasses with sensors during their daily lives for a total of 237 hours. All eating events were labeled as Eating, while the other activities were labeled as NonEating. A sampling rate of 400 Hz was used and we used a window size of 900. For our evaluation, we randomly selected 50,000 samples from the entire data set.

**Fetal Movement Detection [16].** This data is for detecting fetal movements from maternals’ abdomen. The data was measured from 13 participants. There are three classes of data in the dataset. The movements of the fetus are distinguished from those of the mother’s respiration and laughter. MPU 9250 [35] was used as the triaxial accelerometer device and it measured data with a sampling rate of 280 Hz. When processing data, we used a window size of 300 and instances were considered labeled if the window completely contained a single label. As the laughter data was not sufficient for training, we only used respiration and fetal movement data.

**4.1.2 Data Preparation** For models that require images as input, we created spectrograms from the data using the matplotlib framework [33]. Different (*nfft*, *noverlap*) parameters were used for the conversion: (128, 120) for WISDM,

(128, 110) for Goat Activity Recognition, (128, 116) for Roadway Anomaly Detection, and (256, 224) for Eating Detection and Fetal Movement Detection. As described in Section 3.1, each spectrogram was concatenated into one RGB image. Every spectrogram image was resized to  $128 \times 96$ .

The data were randomly split into 70% training data and 30% testing data for each dataset. In our experiments, we selected small samples (e.g., 10 shots) from the training data during fine-tuning to test few-shot applications. The full training data was used to determine the performance upper bound with the Fully-Supervised model.

**4.1.3 Baselines** We evaluate how *IMG2IMU* performs on different types of IMU sensing applications compared with the existing baselines. We list the baseline methods including the state-of-the-art self-supervised learning below:

**RandInit-1D.** We set the ConvolutionalLSTM [53] model as one of our supervised baselines. The weights of the model are randomly initialized, without any pre-trained weights. It accepts 1D waveform data as input. Based on the window size of the datasets, the input and output layers of the model were modified.

**RandInit-2D.** We use ResNet18 [30] model, which has been initialized with random weights, as the supervised baseline using a spectrogram as input.

**ImageNet-Supervised.** A popular vision method in utilizing learned representation is to load weights from the model pre-trained through ImageNet with labels. We include it as a baseline to show the effectiveness of using self-supervised learning for pre-training. It is implemented based on ResNet18, and takes a spectrogram as input.

**TPN-UKB [78].** The TPN [61] framework serves as the baseline for the existing self-supervised learning method for sensing tasks. TPN-UKB [78] used the largest scale of data [20] to generate a pre-trained model for sensing tasks. TPN-UKB is benchmarked as the state-of-the-art baseline in our evaluation, and we used the publicly available code and the pre-trained model provided by the authors of the paper. During the fine-tuning process, we modified the input and output sizes of the last fully connected layer of TPN to fit to the datasets.

**Fully-Supervised.** With the full training dataset, we developed a baseline named Fully-Supervised to set the upper bound for the model using ResNet18 structure with 2D input.

**4.1.4 Training Configurations** For training *IMG2IMU*, we used ResNet18 [30] as the backbone network and trained it using a stochastic gradient descent (SGD) optimizer for both pre-training and fine-tuning. To implement our method, we modified the MoCo [28] framework.

In contrastive learning, four augmentations were used to generate positive views: TranslateX, PermuteX, Hue, and Jitter. Our pre-training data, ImageNet data was resized to match the spectrogram images. Pre-training was conducted using 40 epochs with a learning rate of  $1e^{-6}$  and a batch size of 256. For MoCo hyperparameters, we used a reduced feature dimension of 64 and queue size 4,096 to decrease the computational load.

In the fine-tuning step, we load the model weights pre-trained through contrastive learning and replace the last fully connected layer of ResNet18 with a randomly initialized layer. To preserve the pre-trained weights, we used a two-phase training strategy. In the first phase, we froze the pre-trained layers and only trained the last layer for 50 epochs, with a learning rate of 0.6, which decayed to 0.06 at the 40-th epoch. Next, the whole network is trained for 100 epochs with a learning rate of  $1e^{-4}$ . A batch size of 16 is used for all fine-tuning experiments.

The implementation for RandInit-2D and ImageNet-Supervised follows exactly the same configurations as our system. The models are all based on ResNet18, trained using the SGD optimizer with a learning rate of  $1e^{-4}$ . Instead, RandInit-2D starts training from the random-initialized weights, and ImageNet-Supervised starts training from the weights pre-trained from ImageNet with labels. To load the weight, we used TorchVision’s Multi-Weight Support API [56] that helps directly load the pre-trained weights on ResNet18. RandInit-1D is implemented based on ConvLSTM [53]. We used six convolutional blocks composed of convolutional, batch normalization, and ReLU activation layers, followed by a unidirectional LSTM layer composed of three hidden layers. The input and output sizes of the layers were slightly modified depending on the shape of the input data.

Our implementation for TPN-UKB follows the setting in the paper [78]. They used 1D ResNet as their base network, followed by a fully connected layer for the classification task. To minimize modification, we simply resized the input size of the last classification layer. The pre-trained weights from UK-Biobank dataset [20] were loaded from their public repository.

To make a fair comparison between the baselines, batch size of 16 and training epochs of 150 was used in all experiments. The evaluations were conducted using the PyTorch framework [54] for implementation and trained on the environment with eight NVIDIA TITAN Xp GPUs.

**4.1.5 Metric** The evaluation datasets contain extreme class imbalances. As an example, eating detection data have a sample ratio of 1:14 between eating and non-eating events. We use *macro-averaged F1-score* over classes as our main performance metric, which is robust under class imbalance and thus widely used for imbalanced datasets [55].

## 4.2 Performance on IMU Sensing Tasks

**4.2.1 Overall Results** We conducted an experiment to investigate the performance of the fine-tuned model by *IMG2IMU* against the baselines when only few data are available for IMU sensing applications. In this experiment, we assume that only 10 samples per class are available for training. RandInit-1D and RandInit-2D are used as baselines for comparing with supervised performance using 1D waveform and 2D spectrogram inputs.

Table 3 shows the result. Compared with random initialized models and ImageNet-supervised models, *IMG2IMU* consistently produces high performance. *IMG2IMU* significantly improves the performance of ResNet classification in WISDM, Road Anomaly Detection, and Goat Activity Recognition where randomly initialized ResNet shows worse performance than ConvLSTM. In the WISDM dataset, TPN-UKB shows the best performance as the pre-trained representation fits the human activity recognition tasks. However, *IMG2IMU* is more effective than TPN in tasks from other datasets. While TPN struggled to show generalizability across diverse tasks, *IMG2IMU* generally performed well in a few-shot learning environment using IMU sensor data.

**4.2.2 Evaluating Learned Representations** To evaluate the performance of learned representation, we conducted an experiment based on a linear evaluation protocol [9]. In linear evaluation protocol, the base layers used for the pre-training are frozen except for the last layer, which is replaced with a classification layer. Thus, the base layer operates as a fixed feature extractor based on the learned representation. Here, the randomly initialized baselines, RandInit-1D and RandInit-2D are omitted as their backbone network cannot extract any useful features with the random weights.

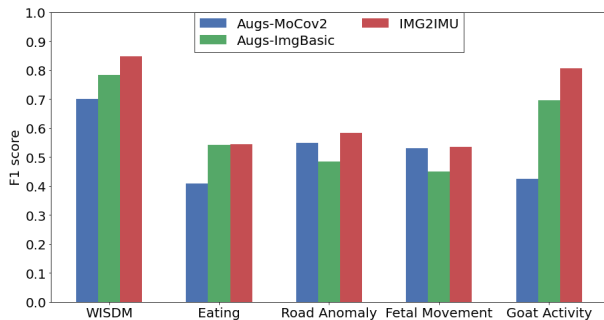
As described in Section 4.1, we trained for 50 epochs during the two-phase fine-tuning process of *IMG2IMU*. We used ImageNet-supervised and TPN-UKB as baselines. We take into account that the baselines all go through pre-training, which enables the use of learned representations. For all settings, a total of 50 epochs are used as the training epoch for a fair comparison.

**Table 3: End-to-end transfer learning performance of *IMG2IMU* in 10-shot learning setting. F1-score is reported as the metric. The highest F1-score ( $\pm 0.01$ ) is shown in bold.**

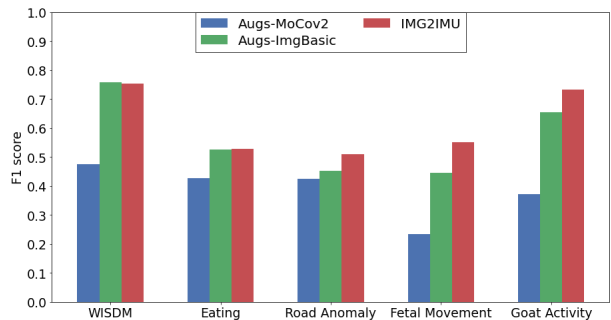
	WISDM	Eating	Road Anomaly	Fetal Movement	Goat Activity
Fully-Supervised	0.9993	0.7024	0.6944	0.6484	0.9769
RandInit-1D	0.6371	0.3482	0.4963	0.5056	0.6335
RandInit-2D	0.6286	0.4700	0.4368	<b>0.5396</b>	0.6194
ImageNet-Supervised	0.7116	<b>0.5367</b>	0.4434	0.4868	0.4898
TPN-UKB	<b>0.8669</b>	0.5150	<b>0.5775</b>	0.5190	0.7473
<i>IMG2IMU</i>	0.8470	<b>0.5434</b>	<b>0.5831</b>	<b>0.5359</b>	<b>0.8075</b>

**Table 4: Linear evaluation performance of *IMG2IMU* in 10-shot learning setting. F1-score is reported as the metric. The highest F1-score is shown in bold.**

	WISDM	Eating	Road Anomaly	Fetal Movement	Goat Activity
ImageNet-Supervised	0.7156	0.5066	0.4009	0.5218	0.6039
TPN-UKB	0.7293	0.4058	0.4671	0.4012	0.6214
<i>IMG2IMU</i>	<b>0.7544</b>	<b>0.5289</b>	<b>0.5106</b>	<b>0.5509</b>	<b>0.7332</b>



(a) End-to-end evaluation.



(b) Linear evaluation.

**Figure 5: Evaluation on the importance of the augmentation choices. We evaluate the F1-score of three pre-trained model with different augmentation set: MoCo v2 augmentations (Augs-MoCov2), superset of *IMG2IMU* with basic augmentations (Augs-ImgBasic), and *IMG2IMU*.**

Table 4 shows that *IMG2IMU* achieved the highest F1-scores for all datasets in the linear evaluation. *IMG2IMU* achieves better results than TPN even in WISDM. It indicates that the learned representation from *IMG2IMU* involves enough knowledge of invariants that should be considered in interpreting sensory tasks, and the representation itself without the fine-tuning is strong enough to work as a feature extractor for sensing applications.

**4.2.3 Effect of Augmentation Set in Contrastive Learning** We evaluate the effectiveness of our augmentation set in contrastive learning for IMU sensing tasks, compared against other combinations of augmentations. In reference to commonly used augmentations [11, 65], we identified two benchmarks. The first baseline is the original augmentation set that is primarily used in MoCo v2 and v3 [11, 12]. The

best-performing augmentation set for MoCo was updated in MoCo v2 and v3. The augmentations involve random crop and resize, random color jittering, random horizontal flipping [75], and applying random gaussian blur to the image [9]. For our experiment, we replicated the augmentation set following the settings of original paper [11] as a baseline and termed it Augs-MoCov2.

To demonstrate that more augmentations do not necessarily improve performance in our setting, we composed a baseline set of widely used image augmentations in addition to the selected augmentations of *IMG2IMU*. Based on a survey on image augmentations [65], we selected the most basic geometric, color space, and random pixel-wise noise injection augmentations to establish the baseline. For geometric augmentations, rotating, cropping out a random region, flipping vertically, and flipping horizontally were used, and for

color space augmentations, random contrast and brightness adjustments, inverting RGB values, and grayscaling were used. The chosen augmentations were merged with the augmentation set of *IMG2IMU*, and we named the baseline as Augs-ImgBasic in our experiment. Note that Augs-ImgBasic is a superset of our augmentations. Our augmentation set, composed of TranslateX, PermutateX, Hue, and Jitter, was compared with the baselines.

Evaluations were conducted in two different manners; end-to-end transfer learning evaluations and linear evaluations. Augs-MoCov2 and Augs-ImgBasic are trained on the setting same as *IMG2IMU*, by replacing the augmentation set used in the contrastive learning. Figure 5 shows the result. Overall, the augmentation set of *IMG2IMU* works the best across various IMU sensing tasks both in the end-to-end transfer learning and the linear evaluation settings. Moreover, although Augs-ImgBasic contains all augmentations that our method has, it performs worse than ours. This result highlights the importance of choosing proper augmentations for knowledge transfer from images to sensors and demonstrates that our augmentation set is effective for diverse IMU sensing tasks.

## 5 Discussion

### 5.1 Augmentations for Sensing

The selection of augmentation types in contrastive learning strongly affects the performance of the fine-tuned model on downstream tasks. *IMG2IMU* utilizes four types of augmentations that benefit contrastive learning for IMU sensing tasks. This augmentation design was derived from the basic invariants that should be considered in sensing applications, referring to the widely accepted sensor data augmentations [71]. While we also attempted other types of image augmentation, such as Brightness and Contrast, to understand whether they are relevant for sensor properties, no other type showed clear correlation. Nevertheless, as there are still numerous invariants present in sensor data, there could still be room for research into other augmentations that could be useful for sensing applications. Several studies have been conducted on the invariants that should be considered for sensor data [2, 43, 62]. More image augmentations can be built upon these works, and could potentially improve the pre-trained model’s performance with *IMG2IMU*.

### 5.2 Task-Specific Augmentation Selection

We defined a fixed set of augmentations based on the commonly known invariants for general IMU sensing tasks. However, for the ideal use of augmentations in contrastive learning, the invariants should be considered more specifically depending on the task, and the augmentation selection should be task-specific by considering the property. For example, in

a system where IMU sensors are attached at fixed positions with a limited range of movement such as static robots, the variability in rotation of the axes will be reduced. This results in less effectiveness of the Hue augmentation as it mainly affects the robustness against the varying sensor rotation. Sensing application developers should consider the invariants in their sensing tasks to compose the augmentation set to be used in *IMG2IMU*. Exploration of more subdivided invariants and their subsequent augmentations for constructing a task-specific augmentation set would be interesting future research.

### 5.3 Effect of Using 2D-transformed Data

To apply the knowledge learned from images, we transform the IMU sensor data into spectrogram images. The design decision was based on the fact that spectrogram is a widely accepted 2D-transformation technique for sensors. While we showed that conversion to spectrograms can benefit a diverse range of sensing tasks when combined with *IMG2IMU*, there could be occasions when spectrograms fail to capture important features. For instance, the spectrogram cannot properly reflect the characteristics of the data when the Fourier transform is performed using an *nfft* parameter that is either too large or too small in our conversion process. In other words, the spectrogram conversion process is sensitive to a few parameters. To address this issue, other types of sensor 2d-transformation methods could be incorporated in addition to spectrograms. It has been reported that other types of 2D representations [34] for sensor data work well as input features for sensory classification tasks. The 2D representations could be used with *IMG2IMU* by designing new types of augmentations that are appropriate for their conversion method.

### 5.4 Converting Image into 1D Format

The performance of *IMG2IMU* is dependent on the conversion method of sensor data. If the converted image from sensor data does not contain enough feature information required for the sensing task, it might lead to the performance drop of *IMG2IMU*. While we adopted the approach of converting sensor data into 2D, to mitigate this problem, we can consider using the sensor data as 1D as-is, but transforming the images from public datasets into 1D for pre-training the model. A simple method that can be used for the conversion would be flattening the image column-wise or row-wise. In this setting, despite the difficulty of training visual features, we can avoid the loss caused by sensor data conversion in fine-tuning. There could be more options for converting image data and useful features could be derived from the converted images for pre-training.

## 6 Conclusion

We presented *IMG2IMU* that utilizes the learned representation from images to IMU sensing tasks. In *IMG2IMU*, we proposed a contrastive learning method that employs image augmentations designed specifically for sensing applications and correlate each augmentation type with sensory properties. Our evaluations demonstrated that *IMG2IMU* improves the performance of few-shot learning on a variety of IMU sensing applications when fine-tuned to the learned representations. *IMG2IMU* showcased how vision knowledge can be effectively applied to IMU sensing tasks. We believe *IMG2IMU* would be especially useful for IMU sensing applications that lack large-scale training data.

## References

- [1] Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee-Pink Tan. 2016. Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.
- [2] Mehmet Arslan, Metehan Guzel, Mehmet Demirci, and Suat Ozdemir. 2019. SMOTE and gaussian noise based sensor data augmentation. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 1–5.
- [3] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. 2021. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3478–3488.
- [4] Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems* 32 (2019).
- [5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
- [6] Manuel Ricardo Carlos, Luis C González, Johan Wahlström, Graciela Ramírez, Fernando Martínez, and George Runger. 2019. How smart-phone accelerometers reveal aggressive Driving Behavior?—The key is the representation. *IEEE Transactions on Intelligent Transportation Systems* 21, 8 (2019), 3377–3387.
- [7] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
- [8] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
- [10] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems* 33 (2020), 22243–22255.
- [11] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).
- [12] Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9640–9649.
- [13] Cristina Cristalli, Nicola Paone, and RM Rodríguez. 2006. Mechanical fault detection of electric motors by laser vibrometer and accelerometer measurements. *Mechanical Systems and Signal Processing* 20, 6 (2006), 1350–1361.
- [14] Jifeng Dai, Kaiming He, and Jian Sun. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3150–3158.
- [15] Omid Dehzangi, Mojtaba Taherisadr, and Raghvendar ChanganVala. 2017. IMU-based gait recognition using convolutional neural networks and multi-sensor fusion. *Sensors* 17, 12 (2017), 2735.
- [16] Upekha Delay, Sajjan Dissanayake, Thoshara Nawarathne, Wishmi Wasalaarachchi, Hetti Arachchi, Sachitha Abeywardhana, Thanushi Withanage, Samitha Gunarathne, Mervyn Parakrama Ekanayake, GMRI Godaliyadda, et al. 2019. Fetal Movement Detection Dataset Recorded Using MPU9250 Tri-Axial Accelerometer. *Mendeley Data* 2

- (2019), 2019.
- [17] Shohreh Deldari, Hao Xue, Aaqib Saeed, Daniel V Smith, and Flora D Salim. 2022. COCOA: Cross Modality Contrastive Learning for Sensor Data. *arXiv preprint arXiv:2208.00467* (2022).
- [18] Analog Devices. 2013. ADXL313. <https://www.analog.com/>.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [20] Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christopher G Owen, et al. 2017. Large scale population assessment of physical activity using wrist worn accelerometers: the UK biobank study. *PLoS one* 12, 2 (2017), e0169649.
- [21] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*. PMLR, 647–655.
- [22] Muhammad Farooq and Edward Sazonov. 2018. Accelerometer-based detection of food intake in free-living individuals. *IEEE sensors journal* 18, 9 (2018), 3752–3758.
- [23] Enrique Garcia-Ceja, Md Zia Uddin, and Jim Torresen. 2018. Classification of recurrence plots’ distance matrices with a convolutional neural network for activity recognition. *Procedia computer science* 130 (2018), 157–163.
- [24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.
- [25] Luis C González, Ricardo Moreno, Hugo Jair Escalante, Fernando Martínez, and Manuel Ricardo Carlos. 2017. Learning roadway surface disruption patterns using the bag of words representation. *IEEE Transactions on Intelligent Transportation Systems* 18, 11 (2017), 2916–2928.
- [26] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. 2020. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 international symposium on wearable computers*. 45–49.
- [27] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.
- [28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.
- [29] Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4918–4927.
- [30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [31] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2013. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM conference on embedded networked sensor systems*. 1–14.
- [32] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. 2016. What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614* (2016).
- [33] John D Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering* 9, 03 (2007), 90–95.
- [34] Taeho Hur, Jaehun Bang, Thien Huynh-The, Jongwon Lee, Jee-In Kim, and Sungyoung Lee. 2018. Iss2Image: A novel signal-encoding technique for CNN-based human activity recognition. *Sensors* 18, 11 (2018), 3910.
- [35] InvenSense. 2017. MPU 9250. <https://invensense.tdk.com/>.
- [36] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Illcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 590–597.
- [37] Mohd Ismifaizul Mohd Ismail, Rudzidatul Akmal Dziauddin, Noor Azurati Ahmad Salleh, Firdaus Muhammad-Sukki, Nurul Aimi Bani, Mohd Azri Mohd Izhar, and Liza Abdul Latiff. 2019. A review of vibration detection methods using accelerometer sensors for water pipeline leakage. *IEEE access* 7 (2019), 51965–51981.
- [38] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. ColloSSL: Collaborative Self-Supervised Learning for Human Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–28.
- [39] Wenchao Jiang and Zhaozheng Yin. 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*. 1307–1310.
- [40] Jacob W Kamminga, Duc V Le, Jan Pieter Meijers, Helena Bisby, Nirvana Meratnia, and Paul JM Havinga. 2018. Robust sensor-orientation-independent feature selection for animal activity recognition on collar tags. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–27.
- [41] Jacob W Kamminga, Nirvana Meratnia, and Paul JM Havinga. 2019. Dataset: Horse movement data and analysis of its potential for activity recognition. In *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*. 22–25.
- [42] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. 2021. Contrastive self-supervised learning for sensor-based human activity recognition. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 1–8.
- [43] Mooseop Kim and Chi Yoon Jeong. 2021. Label-preserving data augmentation for mobile sensor data. *Multidimensional Systems and Signal Processing* 32, 1 (2021), 115–129.
- [44] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1920–1929.
- [45] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* 12, 2 (2011), 74–82.
- [46] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. IMU-Tube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–29.
- [47] Hyeokhyen Kwon, Bingyao Wang, Gregory D Abowd, and Thomas Plötz. 2021. Approaching the real-world: Supporting activity recognition training with virtual imu data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–32.
- [48] Sungjun Kwon, Jeongsu Lee, Gih Sung Chung, and Kwang Suk Park. 2011. Validation of heart rate extraction through an iPhone accelerometer. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 5260–5263.

- [49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [50] Yuan Liu, Ayush Jain, Clara Eng, David H Way, Kang Lee, Peggy Bui, Kimberly Kanada, Guilherme de Oliveira Marinho, Jessica Gallegos, Sara Gabriele, et al. 2020. A deep learning system for differential diagnosis of skin diseases. *Nature medicine* 26, 6 (2020), 900–908.
- [51] Paula Martiskainen, Mikko Järvinen, Jukka-Pekka Skön, Jarkko Tiriainen, Mikko Kolehmainen, and Jaakko Mononen. 2009. Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied animal behaviour science* 119, 1-2 (2009), 32–38.
- [52] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
- [53] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.
- [54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [55] David MW Powers. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061* (2020).
- [56] PyTorch. 2021. TorchVision’s New Multi-Weight Support API. <https://pytorch.org/blog/introducing-torchvision-new-multi-weight-support-api/>.
- [57] Valentin Radu and Maximilian Henne. 2019. Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–21.
- [58] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).
- [59] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. 2016. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)*. IEEE, 71–76.
- [60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Sathesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.
- [61] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.
- [62] Siyu Shao, Pu Wang, and Ruqiang Yan. 2019. Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry* 106 (2019), 85–93.
- [63] Jaemin Shin, Seungjoo Lee, Taesik Gong, Hyungjun Yoon, Hyunchul Roh, Andrea Bianchi, and Sung-Ju Lee. 2022. MyDJ: Sensing Food Intakes with an Attachable on Your Eyeglass Frame. In *CHI Conference on Human Factors in Computing Systems*. 1–17.
- [64] Sungho Shin, Jongwon Kim, Yeonguk Yu, Seongju Lee, and Kyoobin Lee. 2021. Self-Supervised Transfer Learning from Natural Images for Sound Classification. *Applied Sciences* 11, 7 (2021), 3043.
- [65] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.
- [66] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 127–140.
- [67] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. 2020. Exploring contrastive learning in human activity recognition for healthcare. *arXiv preprint arXiv:2011.11542* (2020).
- [68] Inertia Technology. 2017. ProMove mini. <https://inertia-technology.com/>.
- [69] Tensorflow. 2017. Transfer learning and fine-tuning. [https://www.tensorflow.org/tutorials/images/transfer\\_learning](https://www.tensorflow.org/tutorials/images/transfer_learning).
- [70] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems* 33 (2020), 6827–6839.
- [71] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. 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*. 216–220.
- [72] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. 2016. The mobiact dataset: Recognition of activities of daily living using smartphones. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, Vol. 2. SciTePress, 143–151.
- [73] Jinqiang Wang, Tao Zhu, Jingyuan Gan, Huansheng Ning, and Yaping Wan. 2021. Sensor data augmentation with resampling for contrastive learning in human activity recognition. *arXiv preprint arXiv:2109.02054* (2021).
- [74] Weiran Wang, Qingming Tang, and Karen Livescu. 2020. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6889–6893.
- [75] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3733–3742.
- [76] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. 2021. LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 220–233.
- [77] Xu Ye, Guanling Chen, and Yu Cao. 2015. Automatic eating detection using head-mount and wrist-worn accelerometers. In *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. IEEE, 578–581.
- [78] Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, David A Clifton, and Aiden Doherty. 2022. Self-supervised Learning for Human Activity Recognition Using 700,000 Person-days of Wearable Data. *arXiv preprint arXiv:2206.02909* (2022).
- [79] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European conference on computer vision*. Springer, 649–666.
- [80] Runfeng Zhang and Chungping Li. 2015. Motion sequence recognition with multi-sensors using deep convolutional neural network. In *Intelligent Data Analysis and Applications*. Springer, 13–23.
- [81] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and

movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference*

*on computer vision*. 19–27.