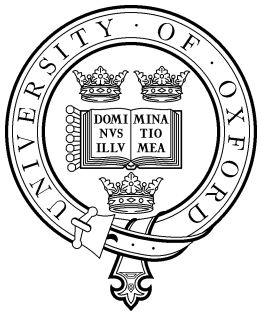
Cross-Modal Learning from Visual Information for Activity Recognition on Inertial Sensors



Catherine Tong

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2021-2022

Acknowledgements

My gratitude for help in writing this thesis runs wide and deep.

As ever, my thanks go first to my supervisor Nic Lane. He brought his wisdom and experience to my support through all important moments during my studies. I thank him especially for taking a leap of faith in a Physics student who knew very little about the world of Computer Science.

At the very beginning of the DPhil programme, I took up an internship at Nokia Bell Labs in Cambridge. It turned out to an invaluable experience that sparked my interest in Ubicomp. I owe thanks to Fahim Kawsar, Akhil Mathur, Sourav Bhattacharya, Alberto Gil Ramos, and Vincent Tseng, who first introduced me to the field. I also would like to thank Danielle Belgrave who was my manager during an internship at Microsoft Research Cambridge, for whose advice I am particular grateful. At Oxford, I spent a considerable amount of time at the Big Data Institute working with Aiden Doherty and his group, namely Shing Chan, Hang Yuan, Rosemary Walmsley and Andrew Creagh, who always made me feel welcomed.

I am also grateful to my friends and family who have been a constant source of support and advice during the past few years, particularly: Milad Alizadeh, Javier Fernandez-Marques, Shyam Tailor, Edgar Liberis, Shuyu Lin, Ronnie Clark, Bo Yang, Yuge Shi, Hyeokhyen Kwon, Harish Haresamudram, Chongyang Wang, Yan Gao, Xinchi Qiu, Priscilla Wong, Emma Rocheteau, Elden Tse, Dongge Han and Lily Liu.

I also thank my mother, without whom my journey in the UK would not have even started.

Finally, I would like to thank William, who has to endure the ups and downs of a PhD without earning one himself. He has powered me through different challenges and this thesis would not have been possible without him.

Abstract

The lack of large-scale, labeled datasets impedes progress in developing robust and generalized predictive models for human activity recognition (HAR) from wearable inertial sensor data. Labeled data is scarce as sensor data collection is expensive, and their annotation is time-consuming and error-prone. As a result, public inertial HAR datasets are small in terms of number of subjects, activity classes, hours of recorded data, and variation in recorded environments. Machine learning models, developed using these small datasets, are effectively blind to the diverse expressions of activities performed by wide-ranging populations in the real world, and progress in wearable inertial sensing is held back by this bottleneck for activity understanding.

But just as Internet-scale text, image and audio data have pushed their respective pattern recognition fields to systems reliable enough for everyday use, easy access to large quantities of data can push forward the field of inertial HAR, and by extension wearable sensing. To this end, this thesis pioneers the idea of exploiting the visual modality as a source domain for cross-modal learning, such that data and knowledge can be transferred across to benefit the target domain of inertial HAR.

This thesis makes three contributions to inertial HAR through cross-modal approaches. First, to overcome the barrier of expensive inertial data collection and annotation, we contribute a novel pipeline that automatically extracts virtual accelerometer data from videos of human activities, which are readily annotated and accessible in large quantities. Second, we propose acquiring transferable representations about activities, from HAR models trained using large quantities of visual data to enrich the development of inertial HAR models. Finally, the third contribution exposes HAR models to the challenging setting of zero-shot learning; we propose mechanisms that leverage cross-modal correspondence to enable inference on previously unseen classes.

Unlike prior approaches, this body of work pushes forward the state of the art in HAR not by exhausting resources concentrated in the inertial domain, but by exploiting an existing, resourceful, intuitive, and informative source, the visual domain. These contributions represent a new line of cross-modal thinking in inertial HAR, and suggest important future directions for inertial-based wearable sensing research.

Contents

1 Introduction **1**

1.1 Research Questions and Contributions 3

1.2 Publications 6

2 Background **8**

2.1 Activity Recognition Using Wearable Inertial Sensors 8

2.1.1 Wearable Inertial Sensor Data 9

2.1.2 Methods for Inertial Activity Recognition 15

2.2 Learning Across Modalities 18

2.2.1 Cross-Modal Data Generation 19

2.2.2 Cross-Modal Knowledge Transfer 22

2.2.3 Zero-Shot Learning 26

2.3 Summary 28

3 Generating Virtual Inertial Data from Videos **29**

3.1 Introduction 29

3.2 IMUTube 31

3.2.1 Overview 31

3.2.2 Motion Estimation for 3D Joints 33

3.2.3 Global Body Tracking in 3D 34

3.2.4 Generating Virtual Sensor Data 38

3.2.5 Distribution Mapping for Virtual Sensor Data 39

3.3 Experiments 39

3.3.1 Establishing Activity Recognition Feasibility 39

3.3.2 Understanding Virtual IMU Data 46

3.3.3 Transfer Learning with Virtual IMU Data 54

3.4 Discussion 58

3.5 Related Work . . . 60
3.6 Summary . . . 61

4 Transferring Knowledge from Visual to Inertial Activity Recognition 62
4.1 Introduction . . . 62
4.2 Knowledge Transfer from Videos to Inertial HAR . . . 63
4.2.1 Problem Formulation . . . 63
4.2.2 Architectural Overview . . . 64
4.2.3 Training CrossHAR . . . 65
4.2.4 Choice of base networks . . . 67
4.3 Experimental Setup . . . 68
4.3.1 Datasets and Tasks . . . 69
4.3.2 Model Implementations . . . 70
4.4 Experiments . . . 71
4.4.1 Activity Recognition Performance on Realworld . . . 71
4.4.2 Are enriched feature representations being learnt? . . . 72
4.4.3 Impact of Training Data Availability . . . 75
4.4.4 Evaluation on HHAR and WISDM . . . 78
4.5 Discussion . . . 79
4.6 Related Work . . . 81
4.7 Summary . . . 82

5 Zero-Shot Learning For Inertial Activity Recognition Using Video Embeddings 83
5.1 Introduction . . . 83
5.2 Proposed Approach . . . 85
5.2.1 Problem Formulation . . . 85
5.2.2 Constructing the Video Semantic Space . . . 86
5.2.3 Projecting IMU Features into Video Semantic Spaces . . . 87
5.2.4 Learning the Projection Function . . . 89
5.2.5 Combining Semantic Spaces . . . 89
5.3 Experimental Setup . . . 90
5.3.1 IMU Data . . . 90
5.3.2 Video Data . . . 91
5.3.3 Baseline Semantic Spaces . . . 92

5.3.4 Model Implementations . . . 93
5.4 Experiments . . . 94
5.4.1 Effectiveness of the Video Semantic Space . . . 94
5.4.2 Scalability of the Video Semantic Space . . . 96
5.4.3 Combining Semantic Spaces . . . 99
5.5 Discussion . . . 101
5.6 Related Work . . . 103
5.7 Summary . . . 104

6 Conclusion **106**
6.1 Summary . . . 106
6.2 Future Work . . . 108

References **111**

A Appendix **131**
A.1 Collaboration Acknowledgments . . . 131
A.2 Train-Test Splits for Zero-Shot Learning . . . 132
A.3 Attribute Space for Zero-Shot Learning . . . 133

List of Figures

1.1 This thesis presents three novel cross-modal learning contributions in inertial HAR. First, we present a solution to the problem of high costs in collecting IMU data, a pipeline that enables video-to-IMU data generation; the generated virtual IMU data is then used for training data of HAR models, which at inference time encounters real IMU data. Second, we introduce a framework that enables knowledge transfer from video-trained models, which improves test-time inertial activity recognition. Third, we utilize video embeddings as an informative auxiliary space to enable zero-shot learning in HAR, so that IMU data belonging to unseen activity classes may be recognized at inference time. 4

2.1 An illustration of projection-based methods for zero-shot learning. During model development, training instances belonging to the seen classes are used to learn the projection function θ, which projects instances from the feature space $\mathcal{X}$ to prototypes in the semantic space $\mathcal{P}$. At inference time, θ is used to project testing instances to the semantic space, where classification is performed using nearest neighbor classifiers or their variants. 27

3.1 The proposed IMUTube system replaces the conventional data recording and annotation protocol (upper left) for developing sensor-based human activity recognition (HAR) systems (upper right). We utilize existing, large-scale video repositories from which we generate virtual IMU data that are then used for training the HAR system (bottom part). 32

3.2 3D joint orientation estimation and pose calibration. The multi-person 2D poses are estimated with *OpenPose* followed by lifting to 3D through *VideoPose3D*. The camera intrinsic parameters are estimated using the *DeepCalib* model. Jointly with the pose and camera related parameters, we calibrate the orientation and translation in the 3D scene for each frame. 34

3.3 3D pose and motion tracking with compensation of the camera motion. The camera motion is estimated through the iterative closest point (ICP) algorithm between subsequent point clouds. Then, calibrated 3D poses per frame are mapped to the location in the entire 3D scene, compensating for camera motion. The calibrated 3D poses from both frames are initially centered in the 3D world origin as the camera follows the cyclists. After incorporating ego-motion information, we can see that two cyclists are moving from right to left, moving closer to each other as in the video (most right figure). 36

3.4 Comparison between virtual and real IMU on the TotalCapture dataset for a user who is performing the "walking" activity. The virtual IMU data is generated by passing a video clip (which is time-synchronous with the real IMU data) through IMUTube with distribution mapping applied. The aligned time series are depicted for sensors placed on the user's left wrist in (a) and left ankle in (b). 48
3.5 Frechet Inception Distance (FID) and confidence interval (shaded area) between virtual IMU and real IMU data distributions after performing distribution mapping of the virtual IMU data with increasing amounts of real IMU samples. The dotted horizontal line is the FID score obtained when the entire real IMU dataset is used for distribution mapping. 52
3.6 Mix2R and R2R performance of a random forest model on 4 different HAR tasks when different amounts of real data per class (in seconds) are used for training. The ratio of virtual data and real data is kept at 1:1 at all datapoints. 53
3.7 Transfer learning vs. amount of real data used for training. 57

4.1 Overview of CrossHAR. An off-the-shelf, pre-trained visual encoder $f(\cdot)$ is present at training time only. The Inertial branch $h(\cdot)$ is trained by predicting correct activity class labels for an input IMU sample, while the Cross-modal branch $g(\cdot)$ is additionally guided by a distillation loss to mirror the outputs of the visual encoder. Together, the two inertial encoders learns complementary embeddings and are kept at inference time to provide rich representations of IMU data for activity recognition. 66
4.2 Normalized confusion matrices on Realworld when training with baseline and CrossHAR models. We show that training with videos results in large reductions in class confusions, with double-digits improvements in some cases. 73
4.3 Visualization of cross-modal transitions, darker lines denote higher similarity (and thus associations): (a) visualizes cross-modal transition probabilities between IMU and video samples in a random batch, with IMU features computed by the IMU-only baseline without any knowledge transfer; Its left half depicts $P^{\text{IMU}\rightarrow\text{Video}}$ and the right $P^{\text{Video}\rightarrow\text{IMU}}$. (b) shows these transition probabilities after knowledge transfer on CrossHAR, which are now grounded within the visual embedding space – IMU samples with different class labels no longer congregate into the same video sample but instead are associated with distinct video samples, suggesting that the inertial features have become more class-discriminative after knowledge transfer. (c) and (d) illustrate a similar case by showing the baseline and CrossHAR transitions on another random batch. 74
4.4 F1 results on Realworld as the amount of video training data increases, expressed as a multiple of the amount of IMU training data. We observe that training with even $0.6\times$ video data brings decent performance gains. We show that CrossHAR can exploit additional video data effectively as the level of performance gains generally increases with the amount of video data. 77
4.5 F1 results on Realworld as the amount of IMU training data increases, as varied by the number of training users. Video-to-IMU data ratio is kept at 10. We observe that IMU data from more than 2 training users is necessary for the effective training of CrossHAR, after which our method consistently outperforms baselines by a large margin. 77

5.1 Overview of our projection-based method for zero-shot learning. The right side illustrates the construction of a video semantic space, which is done by passing video data through a pre-trained I3D model to compute class prototypes. The left side shows the projection of IMU features into the video semantic space, done using a 4-layer MLP. 87
5.2 Example frames from the collected videos belonging to activity classes in the PAMAP2 or DaLiAc datasets. 93
5.3 Per-class F1 scores of all activities encountered in k-fold evaluation on each dataset. ZSL approaches using I3D-both, the best-performing word-based space (i.e. GloVe on DaLiAc and Word2Vec on PAMAP2 and UTD-MHAD), and manual attributes are plotted here for comparison. 97
5.4 Per-class accuracy of our ZSL approach with I3D-logits as its semantic space is plotted for the three datasets, while the number of video clips per classed is increased from 1 to 10 on PAMAP2 and DaLiAc, and from 1 to 32 on UTD-MHAD. We observe an overall upward trend in model accuracy as more video data are used in constructing the video-based semantic space. 99
5.5 Classification accuracy when α is varied from 0 to 1 in steps of 0.01. $\alpha = 0$ represents the case of learning with video-based space only, and $\alpha = 1$ represents the case of word-based space only. 100
5.6 Delta confusion matrices between I3D/I3D+Word2Vec and the Word2Vec on Fold 4 and Fold 3 on the PAMAP2 dataset. The depicted folds are chosen because I3D-logits achieve the best and worst accuracies on them. From left to right: Best fold (I3D-logits), best fold (I3D-logits+Word2Vec), worst fold (I3D-logits), worst fold (I3D-logits+Word2Vec). In each matrix, better and worse performance of video embeddings over word embeddings is indicated in green and red respectively; Therefore, on the main diagonal, a positive number appears green, and elsewhere, a negative number appears green. 102

List of Tables

2.1 Popular publicly available labeled datasets used for evaluating activity recognition on wearable inertial sensor data. An approximate size in hours is provided. Among these datasets, only UTD-MHAD and Realworld provide video recordings in addition to inertial sensor data. Note that [1]the Opportunity dataset additionally provides annotations for low-, mid-, and high-level activities; and [2]HHAR provides approximately 4 Hrs of simultaneous measurement recorded by each of 8 smartphones and 2 smartwatches. 12

2.2 Common activity classes appearing in the inertial HAR datasets listed in Table 2.1. We have grouped these class labels into sedentary, physical, and household activity. We observe that the class labels found in many popular inertial HAR datasets have not expanded greatly beyond those listed in this table. 14

2.3 Popular labeled datasets used for evaluating activity recognition on video data. An approximate size in hours is provided. 16

3.1 Recognition results on the Realworld dataset (8 classes) when training models from real IMU data (R2R), from virtual IMU data (V2R), and from a mixture of both (Mix2R). Wilson score confidence intervals are shown. For Random Forest models, V2R achieves 98% of the R2R F1-score, while a Mix2R setup surpasses R2R by 12%. Quoted in brackets is the difference in F1-scores between each model and its R2R baselines. 41

3.2 Recognition results (mean F1-score) on Opportunity dataset (4 classes) and locomotion activities found in PAMAP2 (8 classes) when using different training data. For Random Forest models, V2R achieves 95% of R2R F1-scores on average, while Mix2R outperforms R2R by 5% on average. 43

3.3 Recognition results (mean F1-score) on PAMAP2 (11-cls.) when using different training data. The Random Forest model trained with virtual IMU data including YouTube videos (for four complex activities) recovered 80% of R2R model performance. 45

3.4 Recognition results (mean F1-score, Random Forest) on the 4-class activity recognition task from Opportunity when using training data from other data sources (top rows) and from Opportunity itself (last row, provided for reference). Without distribution mapping, there is a significant drop in performance when using training data collected under different circumstances than the test case, regardless of whether the IMU data is real (66%) or virtual (64%). This is resolved when distribution mapping is applied. 52

3.5 Recognition results (mean F1-score, Random Forest classifier) on all datasets. Different amounts of virtual IMU data are added to real IMU data, kept constant at 300 seconds per class. 53

3.6 F1 scores of transfer learning with supervised pre-training, when evaluated on different HAR tasks. R2R is the baseline trained on real data from scratch. Transfer learning (TL) results show the performance of the models fine-tuned on real data. . 56
3.7 F1 scores of models learnt through unsupervised pre-training on virtual IMU data, followed by fine-tuning on real IMU data. Note that the R2R* setup here also involves unsupervised pre-training, but on real IMU data. 56

4.1 Statistics of IMU and Video data used in this work. The last column displays m, the video-to-IMU data ratios when employing CrossHAR on RealWorld, WISDM and HHAR. 70
4.2 F1-Scores for CrossHAR and baseline models evaluated on three HAR benchmarks. The absolute differences between each CrossHAR model and IMU-Only baseline (top row) are provided in brackets. 72
4.3 We compare F1 results of models trained with video data from different origins, applied on RealWorld, a dataset that provides both IMU and concurrently recorded video data. Holding the amount of video data constant, paired training with concurrent videos indeed performs the best, though this is also the most costly and unlikely setup due to its need for co-occurring IMU and video data. We show CrossHAR effectively offsets such advantage brought by paired training after using more video data, which can be cheaply added by downloading from online video repositories. We highlight in grey the setting used across all our analyses on Realworld. 77
4.4 F1-Scores for CrossHAR and baseline models evaluated on WISDM and HHAR. The absolute differences between each CrossHAR model and IMU-Only baseline (top row) are provided in brackets. 79

5.1 Comparison of zero-shot learning approaches when different semantic spaces are used. Best results among the individual learned spaces are bolded. Best overall results are colored in blue. 96
5.2 Comparison of zero-shot performance when different word-based and video-based semantic spaces are combined. Best results are bolded. 100

A.1 Train-test split of the PAMAP2 dataset from Wang *et al.* 133
A.2 Train-test split of the DaLiAc dataset. 133
A.3 Train-test split of the UTD-MHAD dataset. 133
A.4 Activity types identified in the PAMAP2 dataset. 134
A.5 Activity types identified in the DaLiAc dataset. 134
A.6 Activity types identified in the UTD-MHAD dataset. 134
A.7 Manual attributes for the PAMAP2 dataset defined by Wang *et al.* 135
A.8 Manual attributes for the DaLiAc dataset. 135
A.9 Manual attributes for the UTD-MHAD dataset. 136

List of Abbreviations

ARC	Activity Recognition Chain
Inertial Data	Data from wearable inertial sensors, most commonly accelerometers and gyroscopes.
IMU	Inertial Measurement Unit. We use IMU as a generic reference to wearable inertial sensor.
Visual Data	Data containing visual information, including image, video, depth, 2D and 3D pose, and optical flow data.
HAR	Human Activity Recognition. As a machine learning task, this refers to the classification of activity labels when given a time window of activity data. We use the terms "activity recognition" and "HAR" interchangeably.
Inertial HAR	Human Activity Recognition using data from wearable inertial sensors. This is often referred to simply as *HAR* by researchers in the Ubiquitous Computing community.
Visual HAR	Human Activity Recognition using Visual Data. This is often referred to simply as *action recognition* by researchers in the Computer Vision community.
ZSL	Zero-Shot Learning. This describes a machine learning model which classifies instances belonging to classes without any labeled training instances.

Chapter 1

Introduction

In the relatively short time span since its inception, wearable sensing has transformed from an exclusive area of pervasive computing research that can only be conducted via delicate specialist instruments (e.g. the Mobile Sensing Platform [1]) to its current ubiquity in the form of miniature sensors embedded in everyday devices – more people than ever are carrying at least a smartphone, smartwatch or other high-tech wearable gadgets, whose sensors continuously collect data as people's lives unfold [2].

Central to wearable sensing is the task of Human Activity Recognition (HAR), which automates the intelligent processing of the collected sensor data into useful information about people's activities. The ability to continuously monitor people in their naturalistic environments promises revolutionary changes to diverse fields and applications. Already some of the most widely anticipated applications relate to healthcare [3–6], psychology [7, 8], fitness and sports [9], and context-aware computing [10, 11]. To researchers in healthcare and psychological sciences, this represents a chance to objectively examine human behaviors unobtrusively and use these observations to answer questions: about the monitoring of personal wellbeing [12], or disease-related state assessment [13, 14] and interventions [15], as well as how and when individuals behave differently under varying circumstances [7]. To developers, these activity data streams can be further integrated with novel services in context-aware computing, enabling interaction experiences that seamlessly adapt to users' needs [16, 17]. Altogether, what unites these efforts is the benefit of an automated means of identifying and recording human activities in their naturalistic environments.

At present, many challenges remain to make this vision a reality. Current activity recognition systems are lacking in the range of activities they are able to detect robustly. Now an

Apple Watch enables the automatic tracking of sleep, gait, and some exercise routines [18]; But what about other activities (e.g. eating, socializing) which comprise a meaningful proportion of our daily lives [19]? The challenge in expanding the breadth of recognizable activities is most often one of *data* – behind every activity recognition system is a machine learning model which requires large amounts of annotated examples to learn, but these examples are scarce. Conventional means of acquiring training data rely heavily on carefully constrained experiments, which are costly to stage and consider a mostly homogeneous population sample. Complicating things further, the learned activity recognition systems will need to be deployed to the general public, meaning they must learn to generalize from a limited training dataset, often collected under contrived controlled environments, and make inferences under diverse usage contexts and user profiles. As a result, although an incremental recognition performance increase has been seen from employing different learning methods for HAR in the last decade, data scarcity inevitably poses a formidable bottleneck to significant progress in wearable activity recognition systems [20].

This thesis proposes that solutions may lie outside the wearable sensing mode of activity recognition by considering other sensing modes where developments have been made for activity recognition. In particular, comparing the capabilities of activity recognition systems from visual data (e.g. videos) and those from wearable sensors reveal vast disparities. The former has developed activity recognition models that can reliably detect activities in greater numbers and diversity [21]. Perhaps the most striking example lies in the number of activity classes that are now being learned by state-of-the-art models – at least 400 for the video domain and under 10 for the wearable inertial domain in most cases. What explains this difference? As stated in [22], computer vision has traditionally been at the forefront of activity recognition, with a large number of research efforts going into the activity and gesture recognition from images and videos, while efforts to recognize activities in unconstrained daily life settings only took off when researchers started shifting their attention to on-body sensors. But importantly visual activity recognition is amongst the many machine learning sub-fields that benefited tremendously from the arrival of deep neural networks, which accord well with readily-available "big data" [23]. Visual activity recognition enjoys a scalable data pipeline brought by the proliferation of online video-sharing platforms, which further propelled community efforts in curating large-scale video activity datasets [24–26]. These favorable data conditions are crucially absent in current developments of activity recognition systems from on-body sensor data, where recruitment-based experiments or crowd-sourcing exercises are painstakingly conducted to collect them.

1.1 Research Questions and Contributions

The juxtaposition of activity recognition from visual data and on-body sensor data highlights the challenge of data scarcity in the latter, which exacerbated further challenges in model development and evaluation. Our thesis is that to overcome the long-standing challenge of data scarcity and go beyond the incremental advances in activity recognition systems, we need to look beyond just the collection of sensor data and consider as an external information source the visual domain – a modality that is rich with data and modeling resources.

We substantiate this thesis by building example activity recognition systems with which we demonstrate the benefits of a reduced need for labeled data collection, performance boosts in activity recognition models, as well as expanded inference scenarios. We identify and motivate three relevant areas and detail the research questions that we seek to address. Figure 1.1 summarizes the contributions of this thesis in response to these research questions.

Question 1 Collecting videos of human activities comes at a fraction of the cost and effort associated with collecting IMU sensor data, with the primary reason being that there are millions of videos in the online public domain that are retrievable with text-based queries, whereas sensor data need to be carefully collected and annotated through recruitment-based participant studies. Creating an equalizer between these two forms of data would accelerate research in activity recognition on on-body sensors and their other sensing applications. We intuit that by leveraging computer vision solutions for human body tracking, one can convert videos of human activities into realistic IMU sensor measurements. Assuming that a video and its converted, virtually-generated, IMU data carry the same activity class label, we may then convert entire video datasets into their virtual IMU data equivalents, and hence be able to develop activity recognition models (operating on real IMU data) with much-reduced data costs. *Can we design a pipeline that, upon input of a video depicting human activities, generates a virtual measurement time series from inertial sensors placed on the human body? To what extent can we replace the real sensor data with the virtually generated sensor data for training activity recognition models? What learning frameworks will allow us to draw benefits from the virtual data for activity recognition?*

In Chapter 3, we present a processing pipeline and a series of validation studies to support our thesis that an automated generation of virtual IMU data from videos can replace the labor-intensive practice of collecting large labeled inertial HAR datasets. Our major contribution is the design and evaluation of IMUTube, a novel automated processing pipeline

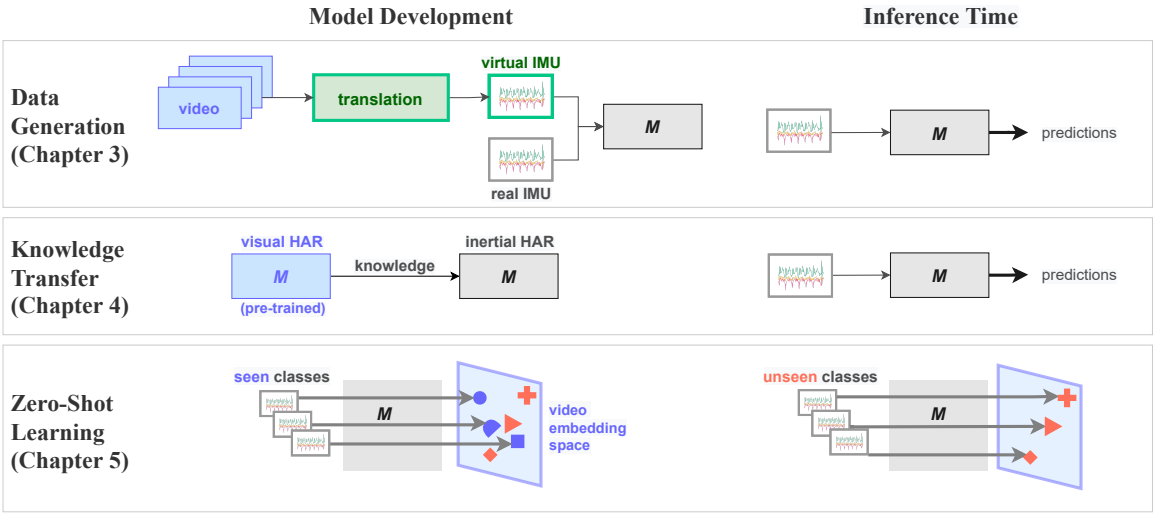


Figure 1.1: This thesis presents three novel cross-modal learning contributions in inertial HAR. First, we present a solution to the problem of high costs in collecting IMU data, a pipeline that enables video-to-IMU data generation; the generated virtual IMU data is then used for training data of HAR models, which at inference time encounters real IMU data. Second, we introduce a framework that enables knowledge transfer from video-trained models, which improves test-time inertial activity recognition. Third, we utilize video embeddings as an informative auxiliary space to enable zero-shot learning in HAR, so that IMU data belonging to unseen activity classes may be recognized at inference time.

that converts videos of human activity into virtual streams of IMU data. This is achieved by integrating a cascade of computer vision and signal processing techniques that visually tracks the human body and estimates its 3D motion trajectory, which is then converted into virtual IMU data. Our validation experiments demonstrate the feasibility of using virtual IMU data to train inertial HAR models, and suggest promising ways to model virtual IMU data, either standalone or in conjunction with real IMU data, to achieve competitive inertial HAR performance.

Question 2 In training neural networks to perform activity recognition on large-scale video datasets, these networks gain *knowledge* that conveys an understanding of human activities and their classification. Given the in-the-wild nature of video datasets, we intuit that such visually-acquired knowledge should provide useful information for complementing the training of inertial HAR models. *Can the knowledge encoded in HAR models from the vision domain be exploited to learn enriched feature representations and improve activity recognition performance in the inertial domain?*

In Chapter 4, we present a framework for transferring cross-modal knowledge from an activity recognition model built using visual data to a model which operates only on inertial data at test time. Our design of the framework addresses data challenges unique to this

cross-modal problem: we focus on visually-extracted motion information to limit the cross-modal information gap, and we assume unpaired and unlabeled video data, which can be easily downloaded from online resources. We conduct a case study on the impact of knowledge transfer using an activity dataset with both inertial and video data. Our results show that the additional knowledge not only provides performance improvements but also shows signs of visually grounding the learned inertial features, which allows for greater model interpretability.

Question 3 When activity recognition models are deployed to the general public, they are bound to encounter activity classes that are of interest to the user but are never previously seen during model development. Zero-shot learning (ZSL) offers a solution to this scenario, by allowing learned models to make inference on unseen classes; The key is to make use of an auxiliary embedding space that relate both seen and unseen classes. In the limited body of previous work investigating the zero-shot learning of inertial HAR, all have used word vectors or hand-crafted attributes to construct such auxiliary spaces – an approach that is directly borrowed from image-based ZSL tasks. Since activity recognition from inertial data rests on a model's understanding of motion patterns, we posit that video embeddings will be a well-suited semantic space candidate due to their motion content. Therefore, our research question is: *How do different auxiliary embedding spaces compare in zero-shot activity recognition performance?* Following that, *Can we construct a video-based embedding space to enable zero-shot inference on on-body inertial data?*

In Chapter 5, we study the use of auxiliary information in zero-shot learning method in inertial HAR. Specifically we compare the recognition performance brought by the use of semantic spaces constructed from hand-crafted attributes, word embeddings, and video embeddings. The use of video embeddings is a novel design in ZSL methods, so we design and implement a strategy to exploit online videos and a state-of-the-art video HAR model to construct an informative semantic space. We conduct experiments to empirically compare the performance of ZSL methods using the different semantic spaces and show that the use of video embeddings can lead to improved performance. We additionally demonstrate that our method of constructing a video-based semantic space enjoys a desirable feature of scalability, as recognition performance was seen to scale with the amount of data used.

1.2 Publications

During my DPhil studies, I have had the opportunity to work on a variety of projects with different collaborators. In particular, the core technical contributions of this thesis are based on the following works. When relevant, the publication venue is listed and * is used to denote equal contribution. Appendix A.1 provides details about my contributions.

Chapter 3

[27] H. Kwon*, C. Tong*, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Plötz. IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 4 Issue 3, Ubicomp'21, 2020.

Chapter 4

[28] C. Tong and N. D. Lane. Empowering IMU-based Activity Recognition Through Cross-Modal Knowledge Transfer from Unpaired Videos. Work in submission, 2022.

Chapter 5

[29] C. Tong*, J. Ge*, and N. D. Lane. Zero-Shot Learning for IMU-Based Activity Recognition Using Video Embeddings. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 5 Issue 4, Ubicomp'22, 2021.

The following lists other publications that I had contributed to during my DPhil studies which are not directly related to this thesis, in chronological order:

[30] V. Tseng, S. Bhattacharya, J. Fernandez-Marques, M. Alizadeh, C. Tong, and N. D. Lane. Deterministic Binary Filters for Convolutional Neural Networks. In *Proceedings of the Twnety-Seventh International Joint Conference on Artificial Intelligence*, IJCAI'18, 2018.

[31] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina, and F. Kawsar. Multimodal Deep Learning for Activity and Context Recognition. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 1 Issue 4, Ubicomp'18, 2018.

[32] C. Tong, M. Craner, M. Vegreville, and N. D. Lane. Tracking Fatigue and Health State in Multiple Sclerosis Patients Using Connected Wellness Devices. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 3 Issue 3, Ubicomp'19, 2019.

[20] C. Tong, S. A. Tailor, and N. D. Lane. Are Accelerometers for Activity Recognition a Dead-end? In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, HotMobile, 2020.

[33] C. Schroeder de Witt*, C. Tong*, V. Zantedeschi, D. De Martini, A. Kalaitzis, M. Chantry, D. Watson-Parris, and P. Biliński. RainBench: Towards Data-Driven Global Precipitation Forecasting from Satellite Imagery. In *the 35th AAAI Conference on Artificial Intelligence*, AAAI'21, 2021.

[34] C. Tong*, E. Rocheteau*, P. Veličković, N. D. Lane, and P. Liò. Predicting Patient Outcomes with Graph Representation Learning. In *Springer Series on "Studies in Computational Intelligence"*, The 5th International Workshop on Health Intelligence, held during AAAI'21, 2021. *Runner-Up for Best Short Paper Award.*

[35] H. Yuan, S. Chan, A. P. Creagh, C. Tong, D. A. Clifton, and A. Doherty. Self-supervised Learning for Human Activity Recognition Using 700,000 Person-days of Wearable Data. Work in submission, 2022.

Chapter 2

Background

In this chapter, we provide readers with a background of the main topics that the rest of the thesis will cover. We will begin by reviewing the Human Activity Recognition (HAR) systems operating on wearable inertial sensor data. This is followed by a review of techniques proposed to accomplish cross-modal learning, which transfers the information gleaned from a source modality to a target modality.

2.1 Activity Recognition Using Wearable Inertial Sensors

In this section, we describe the background of Human Activity Recognition (HAR) from wearable inertial sensors, which we also refer to as "inertial HAR". Our discussion builds upon the *Activity Recognition Chain (ARC)*, an exposition of the domain introduced by Bulling *et al.* in a tutorial text [22]. The ARC breaks down the implementation of HAR systems into five steps: (i) data collection, (ii) data pre-processing, (iii) segmentation, (iv) feature engineering, (v) classification. This section will first discuss aspects of inertial HAR data collection (i.e. step i of the ARC) in Section 2.1.1. We then move on to discuss the rest of the ARC concerning classification model development in Section 2.1.2.

2.1.1 Wearable Inertial Sensor Data

Inertial Activity Recognition, first and foremost, relies on the measurement of human body motion, which is then modeled to disambiguate between different activities. The tracking of human body motion is accomplished using wearable inertial sensors. We define Inertial HAR datasets are datasets that record measurements from wearable inertial sensors, most commonly in the form of accelerometers and usually gyroscopes. Accelerometers measure linear acceleration (measured in ms^{-2} or g), while gyroscopes measure the rate of rotation (in $rads^{-1}$), both give tri-axial measurements. In this thesis, we use the term Inertial Measurement Unit (IMU), a sensing node that assembles accelerometers and gyroscopes, as a generic reference to wearable inertial sensors. Although magnetometers, which measure magnetic field, are also sometimes found in the IMU, magnetometers are often not included in inertial HAR datasets and not used as an input to inertial HAR models [36], so we will not consider them further. We refer interested readers to [37] for an extensive review of modern accelerometers and gyroscopes.

Considering these inertial measurements together with annotated activity class labels, they form a natural starting point for the development of supervised learning algorithms [38] for activity recognition. However, building a quality annotated dataset of inertial sensor data is challenging since the data collection process must involve human subjects, and ground truth activity labels are difficult to obtain [39]. In the following, we delve into key issues surrounding the collection of wearable inertial sensor data.

2.1.1.1 Practical Considerations in Data Collection

To collect wearable inertial data for building HAR systems, human subjects are recruited to participate in a data collection study. Decisions made in the data collection process have implications on the produced dataset as well as the eventual capabilities of the recognition system. We discuss a number of these practical considerations below.

Scripted Activities Many studies collect data by asking participants to follow a script of these activities while wearing inertial sensors [40–43], this usually allows researchers to have greater control over the resources required for running the study, the captured behaviors as well as the resulting class distribution of the datasets. Participants are often given the freedom to carry out the prompted activities, and a less specific script may lead to a greater diversity of captured behaviors, which has been attributed to be beneficial for downstream

HAR model performance [44]. Alternatively, activity data collection can be conducted in other ways to capture more naturalistic free-living behaviors, common examples include asking participants to provide their smart device data with self-report annotations [45, 46], or to perform specific activities through mobile app notifications [47]. In general, the data collection strategy should facilitate the capturing of a diverse range of activity expressions, while considering what is feasible given the available time and budgets, as well as the application scenarios of the developed HAR system.

Sensor Placement Wearable inertial sensors are placed on strategic locations of the human body during a data collection study. As the sensors are constrained to move along with the designated body location, they can capture its localized motion information [48]. Determining the locations for sensor placement is important, as different placements and orientations may capture significantly different motion information, and impact the performance of the HAR system. Motion information near the head, arm, wrist, waist, thigh and shin has been found to be relevant for activity recognition [49]. Some studies have collected data from subjects wearing multiple sensors in different body locations simultaneously [50], while many recent studies have limited placements to up to two locations that reflect common placements of commercial smart devices (e.g. a smartwatch on the wrist and a phone in the pocket [51]). It is also important to consider the levels of user comfort, social acceptance, and study compliance that are associated with different placement locations [52].

Sensing Device Inertial sensor measurements may be recorded using a standalone sensor, sensors within an IMU, or those embedded in mobile consumer device platforms, e.g. smart phones and watches. Heterogeneities among sensing platforms contribute to differences in the collected data, which is the subject of study in [42]. Stisten *et al.* summarized the main sensing heterogeneities among accelerometer-based HAR systems as sensor biases, sampling rate heterogeneity, and sampling rate instability [42]; The choice of sensing devices for data collection should therefore take these factors into account. For example, if highly accurate and granular inertial measurements are required, one might collect data using standalone IMU sensors instead of those embedded in commercial devices, which Sisten *et al.* have found to exhibit larger sensor biases (e.g. poorly calibrated with limited granularity). Again these factors should be considered in conjunction with the intended usage scenario of the HAR system as well as perceived user acceptance, for example, participants' willingness to use different sensing platforms may change depending on study duration, environment, and placement locations.

Annotation Protocol Inertial sensor data is difficult, and sometimes impossible, to interpret and annotate [39]. To facilitate the annotation process, simultaneous streams of other sensor modalities are often employed. The simplest setup is to perform each activity in distinct time intervals, such that class labels can simply be assigned according to when each activity starts and ends [43, 50]. Another common approach is to take video recordings, which can be easily set up in an indoor instrumented environment [40]. However, it is often challenging to synchronize both visual and inertial sensor streams for labeling [53], and annotators may need to undergo annotation training to ensure label uniformity and reduce bias [54]. Methods requiring direct input from the participants (e.g. experience sampling [55, 56]) are common amongst in-the-wild data collection studies since researchers cannot be present to record activity labels. Other modalities, such as voice recording from the users [57] and wearable camera streams [54] have also been analyzed retrospectively to obtain activity annotations.

2.1.1.2 Inertial HAR Datasets

Ever since Bao and Intille's pioneering inertial HAR study in 2004 [59], many inertial HAR datasets have been released to the public domain. Table 2.1 depicts a subset of these datasets which are commonly used for evaluating inertial HAR models and will also be referenced in later chapters of this thesis. A description of each dataset is provided below.

Opportunity The Opportunity Activity Recognition dataset [40] is one of the most well-known dataset for inertial HAR evaluation. The dataset provides inertial data from 4 participants performing pre-defined home activities in a laboratory simulating a studio flat. Each participant wore an instrumented motion jacket and shoes, and a simultaneous video recording was used for subsequent activity annotation on 4 different tracks: one is modes of locomotion (e.g. "standing"), two other tracks describe the action of the hand on an object (e.g. "open fridge"), and the fourth track indicates the high-level activities (e.g. "prepare sandwich"). It is common to evaluate inertial HAR models using the 4-class locomotion annotations, which also presents a class imbalance problem ("walking" and "standing" form the majority and "sitting" and "lying" are the minority classes).

PAMAP2 The PAMAP2 Physical Activity Monitoring dataset [41] was collected to provide a benchmark dataset for studying physical activity monitoring from inertial sensors. PAMAP2 provides inertial data from 9 participants over 18 activity classes. While wearing

Dataset	Size	Classes	Subjects	Body Positions	Environment	Video	Sensing Device	Sampling Frequency	Year
Opportunity [40]	7 Hrs	4[1]	4	19	Lab		IMU sensor node	30 Hz	2011
PAMAP2 [41]	8 Hrs	18	9	3	varying		IMU sensor node	100 Hz	2012
DaLiAc [43]	43 Hrs	13	19	4	Lab		IMU sensor node	204.8 Hz	2013
HHAR [42]	40 Hrs[2]	6	9	3	Lab		Smart Phone & Watch	$50 \sim 200$ Hz	2015
UTD-MHAD [58]	< 1 Hr	27	8	2	Lab	✓	IMU sensor node	50 Hz	2015
Realworld [50]	18 Hrs	8	15	7	varying	✓	Smart Phone & Watch	50 Hz	2016
WISDM [51]	60 Hrs	18	51	2	Lab		Smart Phone & Watch	20 Hz	2019

Table 2.1: Popular publicly available labeled datasets used for evaluating activity recognition on wearable inertial sensor data. An approximate size in hours is provided. Among these datasets, only UTD-MHAD and Realworld provide video recordings in addition to inertial sensor data. Note that [1]the Opportunity dataset additionally provides annotations for low-, mid-, and high-level activities; and [2]HHAR provides approximately 4 Hrs of simultaneous measurement recorded by each of 8 smartphones and 2 smartwatches.

IMU sensors on the head, chest, and ankle, the participants were asked to perform 12 activities following a pre-defined protocol, after which they may perform 6 optional activities. This data collection protocol, compounded with missing data due to data transfer issues, resulted in PAMAP2 having an imbalanced data distribution across participants and activity classes. For example, none of the 18 activity classes have been successfully recorded by all 9 participants. As a result, different subsets of the PAMAP2 dataset have been used for evaluating inertial HAR models in previous works (e.g. data from 4 classes are evaluated in [60], 8 classes in [27, 35], and 12 classes in [60–62]).

DaLiAc The DaLiAc (Daily Life Activities) dataset [43] was a database for the classification of daily life activities using inertial data. DaLiAc provides inertial data from 19 participants who all performed 13 activities while wearing 4 sensor nodes, placed on their hip, chest, wrist, and ankle. During the data collection process, a custom-made Android-based labeling App was used to indicate the start and end time of each activity.

HHAR The Heterogeneity Activity Recognition dataset [42] was collected to investigate the impact of sensing heterogeneities on inertial HAR performance, with a focus on the variations between different sensors, devices, and workloads. HHAR provides data from 9 participants performing 6 activities. Each participant was instrumented with 8 smartphones around the waist, and 4 smartwatches on their arms, each recording simultaneously at a different sampling rate.

UTD-MHAD The UTD Multimodal Human Action Dataset (UTD-MHAD) [58] is very different from the rest of the datasets described so far because it only contains short actions

(e.g. *"swipe left"*, *"arm cross"*) and it was collected for the purpose of multi-modal action recognition instead of inertial HAR. Therefore, apart from inertial data, UTD-MHAD also provides time-synchronized RGB video and depth data for 8 subjects. For inertial data collection, an IMU sensor was worn on the participant's wrist or thigh depending on whether the action involved arms or legs. There are 8 subjects in total, and each subject was asked to repeat each of the 27 actions 4 times in a laboratory setting.

Realworld The Realworld dataset [50] was collected to implement inertial HAR approaches in a real-world setting. The dataset provides inertial data from 15 participants who perform 8 pre-defined activities under realistic conditions. For example, when collecting "walking" data, the participants were free to move through the city and forest areas, instead of a conventional laboratory setting. The dataset was also collected to investigate the problem of on-body position detection, so simultaneously measurements were taken from 7 body positions (chest, forearm, head, shin, thigh, upper arm, and waist) using commercial smartphones and smartwatches. Additionally, non-time-synchronous video recordings of the participants performing each activity are released publicly alongside the inertial data. As the Realworld dataset uniquely contains both inertial and video data of human activities, it is of particular relevance to this thesis.

WISDM The Wireless Sensor Data Mining (WISDM) Smartphone and Smartwatch Activity and Biometrics Dataset [51] was collected to explore activity recognition using inertial sensors from commercial mobile devices. WISDM provides data from 51 participants who each performed 18 tasks for 3 minutes each. Each participant wore a smartwatch on their hand and placed a smartphone in their pocket. To examine the feasibility of using wrist-worn smartwatch data for activity recognition, the activities collected in this dataset span a more diverse range of hand-based activities, such as "eating pasta", "brushing teeth" and writing", which is uncommon in other inertial HAR datasets.

Table 2.2 summarizes the most common activities found in the described datasets, which suggests that a great amount of inertial data recorded so far have been concentrated in a narrow spectrum of human activities, in the form of sedentary activities (e.g. "sitting"), physical activities (e.g. "walking") and a select few household activities (e.g. "vacuuming"). Besides the datasets listed above, it is true that a great number of separate attempts have been made at collecting inertial sensor data [45, 63], as well as efforts to collect data for specific activity types, such as fitness exercise [64–66], sports [67], eating behaviors in [68], falls in [69], industrial activities in [70, 71], to name but a few. However, the difficulty in harmonizing the data collected from these varying data collection efforts highlights

Grouping	Activity	Datasets
Sedentary Activity	Sitting	Opportunity [40], PAMAP2 [41], DaLiAc [43], HHAR [42], Realworld [50], WISDM [51]
	Standing	Opportunity [40], PAMAP2 [41], DaLiAc [43], HHAR [42], Realworld [50], WISDM [51]
	Lying	Opportunity [40], PAMAP2 [41], DaLiAc [43], Realworld [50]
Physical Activity	Walking	Opportunity [40], PAMAP2 [41], DaLiAc [43], HHAR [42], Realworld [50], WISDM [51]
	Stairs	PAMAP2 [41], DaLiAc [43], HHAR [42], Realworld [50], WISDM [51]
	Running	PAMAP2 [41], DaLiAc [43], Realworld [50], WISDM [51]
	Jumping	PAMAP2 [41], DaLiAc [43], Realworld [50]
	Cycling	PAMAP2 [41], DaLiAc [43], HHAR [42]
Household Activity	Vacuuming	PAMAP2 [41], DaLiAc [43]
	Folding Clothes	PAMAP2 [41], WISDM [51]

Table 2.2: Common activity classes appearing in the inertial HAR datasets listed in Table 2.1. We have grouped these class labels into sedentary, physical, and household activity. We observe that the class labels found in many popular inertial HAR datasets have not expanded greatly beyond those listed in this table.

an issue of *dataset heterogeneity* that is prevalent in the area of inertial HAR. As discussed in Section 2.1.1.1, many decisions in data collection affect the resulting dataset, and most datasets have been collected using a different sensing device, under a very different data collection strategy, annotation protocol, and labeling schemes. Combining the data from these varying datasets is non-trivial, and the current predominant approach is to consider each dataset separately. Therefore, the problem of annotated data scarcity in inertial HAR remains.

As a comparison to inertial HAR datasets over a similar time span, we list popular visual HAR datasets in Table 2.3. Thanks to the relative ease in data collection, it is apparent that visual HAR datasets have become orders of magnitude larger in size, the diversity of people and activities covered. Throughout this thesis, we extensively refer to the Kinetics dataset, which is detailed as follows.

Kinetics Kinetics [24] is a popular benchmark for vision-based action recognition models and it is by far the largest of its kind. The Kinetics database has been continually expanded in over the years with top-up datasets [25, 26], an operation that is difficult to replicate in inertial HAR datasets due to sensing heterogeneities. Its initial release, Kinetics-400, already provides ≥ 400 RGB video clips for each of 400 classes, giving a total of 306,245 10-second clips (~ 851 hours) [24]. The video clips form a good representation of in-the-wild activities, since each clip is taken from a separate YouTube video and most likely depicts a different human subject. Kinetics span a much wider range of human activities

than those contained in any single inertial HAR dataset, providing annotations relating to arts and crafts, athletics and sports, gardening, animal interactions, dancing, martial arts, playing musical instruments, etc. The dataset also provides fine-grained distinctions for activities typically grouped as one class in inertial HAR settings; a case in point is the "cycling" activity in inertial datasets (e.g. PAMAP2 [41]), which are can be found separately "riding a bike", "riding mountain bike", "biking through snow" in Kinetics-400.

2.1.2 Methods for Inertial Activity Recognition

Following the collection and annotation of activity data, we now discuss the subsequent steps involved in developing an inertial HAR model based on supervised machine learning, i.e. models which learn from example data and their associated annotations [38].

Data pre-processing and segmentation With the acquired raw inertial measurements, it is typical to pre-process and transform them into an appropriate form for machine learning system ingestion. If multiple sensors are involved, these transformations often involve time-synchronizing and concatenating the data, as well as performing temporal re-sampling to ensure that the sampling rate is consistent between modalities. This is commonly followed by sliding-window segmentation of the recorded time series, which is simple to implement compared to other approaches like energy-based [53] or semantic segmentation [22]. Under the sliding window approach, each signal stream is segmented into fixed-length time windows, also known as frames; an overlap of 50% between frames is also commonly used. Overall, these transformations ensure that the data dimensions are standardized for the latter modeling steps. A resulting dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ now provides N instances, with x being of dimensions $t \times d$ (t is the number of time steps in each frame and d is the sensor measurement dimension), and y indicating the corresponding activity class label of the frame. Throughout this thesis, we refer to each frame and its corresponding class label as an individual labeled sample or instance, and consider frame-based model learning and evaluation.

Feature Engineering and Classification Early works in inertial HAR rely heavily on feature engineering to extract distinguishable features for human activities, which are then modeled using classical Machine Leaning (ML) methods. The goal of feature engineering is to extract a descriptive vector from each window of raw sensor measurement, such that motion information useful for disambiguating activities is accentuated. Time-domain

Dataset	Size	Classes	Clips	Resource	Video	Year
HMDB51 [72]	2 Hrs	51	6,766	Movies, YouTube	✓	2011
UCF-101 [73]	27 Hrs	101	13,320	YouTube	✓	2012
ActivityNet [74]	849 Hrs	203	27,801	YouTube	✓	2015
Charades [75]	82 Hrs	157	9,849	Crowd-sourced	✓	2016
Kinetics-400 [24]	850 Hrs	400	306,245	YouTube	✓	2017
AVA [76]	107 Hrs	80	430	YouTube	✓	2018
STAIRS [77]	142 Hrs	100	102,462	YouTube, crowd-sourced	✓	2018
Kinetics-600 [25]	1,376 Hrs	600	495,547	YouTube	✓	2018
Kinetics-700 [26]	1,806 Hrs	700	650,317	YouTube	✓	2019

Table 2.3: Popular labeled datasets used for evaluating activity recognition on video data. An approximate size in hours is provided.

features are often given by extracting statistics from the raw inertial signal over each degree of freedom (e.g. mean x-acceleration) or computing the correlations between degrees of freedom (e.g. cross-correlation between acceleration axes), whereas frequency-domain features are computed after applying a Fast Fourier Transform (FFT) over the input signal sequence. It was also suggested that features based on the Empirical Cumulative Distribution Function (ECDF) of raw data present a concise feature representation that demonstrates good performance for inertial HAR tasks [78]. Finally, the extracted features and their corresponding labels can be used to train classification algorithms. The early work by Bao and Intill [59] used Decision Tree classifiers, and ensemble models of trees such as Random Forest (RF) [79] are popular as they offer a powerful non-parametric discriminative method for classification [80]. In scenarios where long-term recordings are of interest [81], the temporal order of activities may be further modeled using a Hidden Markov Model (HMM) [82]; the use of RF predictions as emission models in HMM was explored in [54, 83, 84] in order to capture the temporal regularities and smoothness of activity sequences.

Deep learning approaches The last decade has seen the widespread application of Deep Learning (DL) methods in inertial HAR tasks, beginning with early investigations by [13, 85, 86]. One advantage offered by deep learning is the combination of feature representation learning and classification into a single objective, which circumvents the time-consuming stage of manual feature engineering [87]. The standard approach for building inertial HAR models involves specifying a neural network architecture, whose weights are iteratively updated via stochastic gradient descent to minimize an error (loss) function [88]. Since inertial HAR is a classification task, the loss function is typically defined as the cross-entropy function between model predictions and the ground truth activity classes.

Popular architectures are variants of Convolutional Neural Networks (CNNs) [89–93], Recurrent Neural networks (RNNs) [94], in particular Long Shot Term Memory (LSTM) networks [62], or a hybrid structure of convolutional and recurrent networks [86, 95–99]. We refer readers to [100] for a comprehensive survey on deep learning methods developed for inertial HAR.

Model evaluation Due to the high variance amongst individuals in how they perform activities, standard evaluation of an inertial HAR model is performed on test subjects who are *unseen* during model training, such that the reported results will give an indication on the model's generalization performance to the general population. As such, datasets are commonly split by *subjects* into training, validation, and testing sets, either in a hold-out scheme or k-fold cross-validation setting. Given limited dataset sizes, a subject-wise split may not always be possible and alternate dataset splits have been used to evaluate different application scenarios, for example, previous works have evaluated inertial HAR models on unseen recording sessions [94], sensor placement locations [61], and activity classes [101]. As for performance metrics, F-measures are very popular as they concisely summarize recognition performance based on both negative and positive predictions. In particular, mean $F1$ scores, also known as macro $F1$ scores, are widely used as they are considered a reasonable evaluation strategy for imbalanced datasets [39]. For a N_C-class classification task, the macro-F1 score is defined as follows [102]:

$$\text{Macro F1} := \frac{1}{N_C}\sum_{k=1}^{N_C}\Big(2 \times \frac{\text{precision}_k \times \text{recall}_k}{\text{precision}_k + \text{recall}_k}\Big) \tag{2.1}$$

The reader should note that there has so far not been a uniform approach in evaluating inertial activity recognition models [100, 103]. For the same dataset, different modes of evaluation may be of interest to different researchers, whose decisions in data pre-processing, segmentation, and dataset splits all influence the reported results. In this thesis, as is common practice in inertial HAR, we fix the evaluation protocol within the context of each technical work (i.e. within each chapter) and define a suitably competitive baseline to facilitate performance comparison.

2.2 Learning Across Modalities

Central to the methods proposed in this thesis is the idea of learning *across* modalities, where information from a source modality A is brought over to a target modality B in order to benefit learning tasks operating on B. We refer to this setup as Cross-Modal Learning, of which we seek to give an overview in this section. By maximally exploiting existing data and models (from the source modality) in lieu of staging new data collections, cross-modal learning is particularly useful for scenarios where there are large gaps in the amounts of data and/or data collection costs in the source vs. target modalities. As a result, cross-modal learning lends itself nicely to our thesis of exploiting visual modalities to benefit activity recognition models operating on inertial modalities.

We have identified three ways that encompass notions of cross-modal learning, namely *data generation*, *knowledge transfer* and *zero-shot learning*. In this section, we review the key ideas and background behind these cross-modal approaches. In later chapters of the thesis, we will build upon these ideas towards three distinct, novel technical methods for leveraging the visual domain to advance the research frontier of inertial HAR, forming the core contributions of this thesis.

Many previous works in inertial HAR have investigated machine learning techniques for optimally fusing data from different sensing modalities [31, 66, 86, 96, 104–106]; We distinguish such a setup, which is generally referred to as "multi-modal learning", from the "cross-modal learning" setup with which this thesis is concerned. We highlight the key differences between the two notions. First, the main research goal of multi-modal learning is the *fusion* of complementary information from multiple modalities, such that the learned representations of the final model can be improved over those learned from any single modality [107]. This is different from the goal of cross-modal learning, which specifies a directional *transfer* of information from source to target modality. Second, multi-modal learning generally assumes the availability of *co-occurring* multi-modal observations during both model development and inference time; In contrast, samples from the target modality are not required at the inference time of cross-modal setups. For further details on multi-modal learning, we refer the reader an extensive survey of the general landscape of multi-modal learning in [108], as well as a study of multi-modal learning in sensor-based activity and context recognition [31], to which the author of this thesis has also contributed.

2.2.1 Cross-Modal Data Generation

Cross-modal data generation refers to the generation of data of a target modality B through some transformation of data from a source modality A. In this section, we will begin by discussing an overview of cross-modal data generation and common adopted data-driven approaches. Afterward, we will proceed to discuss the specific area of cross-modal generation of wearable inertial sensor data.

2.2.1.1 Overview of Cross-Modal Data Generation

When data can be obtained much more cheaply with a modality different from the target modality, data collection costs in the latter may be dramatically reduced if one can easily convert data from one modality to another. Such is one motivation behind cross-modal data generation, where a mapping is used to translate data samples from modalities A to B. Given a high-fidelity mapping that preserves data labels, a large labeled dataset already collected in the source modality can be readily converted into a large labeled dataset of *virtual* data in the target modality. This virtual annotated dataset can then be used to replace or extend any existing dataset in training models, so as to increase its generalization capability.

We formalize this data translation task as follows. Let $x_A \in \mathcal{R}^{d_A \times 1}$ as a vector of data from the source modality A, and $x_B \in \mathcal{R}^{d_B \times 1}$ of data of the target modality B. The source-to-target mapping is applied onto x_A such that its output resembles x_B:

$$\Psi_{A \to B} : x_A \to \hat{x}_B \approx x_B \tag{2.2}$$

In many applications of cross-modal data generation (e.g. image to text), the relationship between A and B is complex and hard to specify manually. This scenario calls for learning the mapping $\Psi_{A \to B}$ in a data-driven manner, where the typical approach is for practitioners to design a neural network architecture that encodes inductive biases about the translation task, and learns its weights through minimizing the difference between $\hat{x}_B$ and x_B. On the other hand, if the relationship between A and B is well-understood, then the required mapping may be manually defined using domain expertise. This is the predominant approach taken in existing works that generate virtual IMU data by translating information from another modality.

2.2.1.2 Applications of Cross-Modal Data Generation

Many examples of learning cross-modal mapping can already be found as well-studied, standalone tasks in different areas of machine learning, typically carried out via deep neural networks. Models for image captioning essentially provide an image-to-text mapping, human pose tracking learns a mapping from RGB images to pose data [109], while automatic speech recognition (ASR) [110] and text-to-speech (TTS) models [111] learn mappings that translate between audio and text data. Pre-trained models from these tasks can then be exploited to generate virtual datasets in cases where target modality data are costly to collect [112, 113]. An example application is found in pose-based activity recognition tasks, which conventionally use human pose data recorded by specialist instruments (e.g. Microsoft Kinect v2) as input. To increase the quantity and variation in recorded poses, Yan et al. [113] translated the Kinetics dataset [24] from its original RGB video data to 2D pose data and created the Kinetics-Skeleton dataset, which has since become a popular benchmark for both model training and evaluation for pose-based HAR [114–116].

The aforementioned examples of cross-modal data translation re-purpose existing pre-trained models for virtual data generation in the target modality. Different from these works, medical imaging is an active area of research where cross-modal data generation models are trained specifically for the purpose of lowering data collection costs [117–120], as the ability to convert data recorded using a variety of imaging techniques (e.g. X-ray, CT, MRI, PET images) can yield significant benefits to training data size and diversity. These methods usually employ Generative Adversarial Networks (GANs) which learn transformation to generate realistic data of modality B conditioned on input data from modality A. However, there is a limited modality gap between these samples, since both A and B are limited to visual modalities, and typically only samples capturing the same anatomy are considered (e.g. cardiac CT and cardiac MR images). It is also common to further pre-process the samples (e.g. through view alignment) such that the source and target samples are similar structurally to aid model training [117]; Together, these manipulations contribute to lowering the difficulty in cross-modal generation in medical imaging.

Whilst it is true that using a learned model like GANs could give rise to verisimilar virtual data, the quality of learning also depends on the complexity of the mapping, as well as whether large amounts of data are available from both source and target modalities. The latter is difficult to guarantee if data collection costs in the target modality are high. As such, the time and data resources involved can potentially make learning the mapping function very costly. Further, physical constraints which should be obeyed in the modality

translation mapping may not be picked up by the learned mapping if there is insufficient training data or the model is too weak.

2.2.1.3 Wearable Inertial Data Generation

We now discuss cross-modal data generation in the context of wearable inertial data. Throughout this thesis, we refer to the wearable inertia data generated through cross-modal generation as "virtual" data, in contrast to the "real" data recorded by directly measuring motion information by placing inertial sensors on human subjects.

The predominant approach taken in virtual inertial data generation relies upon domain expertise to hand-craft the translation relationship between the source and target modalities, such that the required mapping may be manually defined using mathematical and physics knowledge. The key idea is to select a source modality where one can obtain information about human body motion trajectory, after which one may apply reasonable assumptions and domain knowledge (e.g. forward kinematics) to simulate inertial sensor measurements. For example, if the source modality provides a time series of the position coordinates of a wearable accelerometer, then one can approximate its acceleration measurements via finite differences of positions coordinates as follows [121]:

$$\boldsymbol{a}_t = \frac{\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t+1} - 2\boldsymbol{x}_t}{\Delta t^2} \tag{2.3}$$

where $\boldsymbol{x}_t$ is the position coordinate of a virtual inertial sensor at time t, and the time interval between two consecutive frames is Δt.

Previous works mainly explored two types of source modality for inertial sensor data generation:

Motion Capture (Mocap) The most common form of data translation into wearable inertial data employs Optical Motion Capture (Mocap) data as the source modality [122–126]. Mocap data record high-precision motion trajectories by tracking the motion of markers placed on human subjects, and they are typically recorded indoors with an expensive infrastructure provided by commercial systems such as Vicon. A number of pre-existing Mocap data are available in the public domain through large-scale databases, including the CMU motion capture corpus [127] and AMASS [128]. Alternatively, it may suffice to manually design the human motion trajectories in lieu of using Mocap data for simple

motions (e.g. walking). Equation 2.3 underlies the key principle utilized in most recent works which extract virtual IMU data from mocap data to training machine learning models [124, 126, 129].

Animation Alternatively, for very simple motions the motion trajectory might also be hand-crafted and defined directly using Virtual Reality (VR) game development environments like Unity [130]. Kang *et al.* proposed a 3D human model on computer graphics software and simulating human activities [131]. The virtual inertial sensor data is extracted from the simulated human motion, and subsequently used to train the recognition models. However, the cost of manually designing realistic and more complex human activities may remain a bottleneck. Therefore, such methods typically only explore simple gestures and locomotion-based activities (e.g. walking) [131].

To facilitate experimentation in inertial sensing applications, Young et al released IMUSim as an open-sourced simulation tool [122]. IMUSim allows for the generation of realistic inertial measurements from motion trajectories, which may be manually defined through human animation or measured from motion capture systems. Given a motion trajectory, IMUSim approximates a continuous trajectory model of the body motion, which is then sampled to give sensor readings from accelerometers and gyroscopes; these sensor readings are further made more realistic by incorporating real-world sensor noises, including non-ideal sensors, timing factors, and magnetic field distortions. Through verification against simultaneous real sensor data, the authors showed that IMUSim, when provided with high-quality motion capture data together and known IMU sensor characteristics, demonstrates a good match between simulated and real IMU data for the task of full-body motion capture using on-body IMU sensor networks.

2.2.2 Cross-Modal Knowledge Transfer

It is commonly understood that models developed using limited training data are prone to overfitting. Learning with additional *knowledge* is a natural solution to this problem – the additional knowledge may come in the form of contextual data, common sense concepts represented by knowledge graphs, knowledge of a learned neural network, etc. In cross-modal knowledge transfer, we are interested in supplying additional knowledge from source modality A, with which a model operating on target modality B is learned. This problem can be solved using the framework of *knowledge distillation*, which enables fast knowledge transfer from one neural network to another. In the following, we first describe

a vanilla knowledge distillation framework, followed by notable extension works for cross-modal knowledge transfer.

2.2.2.1 Knowledge Distillation

The original motivations of knowledge distillation (KD) stem from model compression [132]: Resource constraints at inference time call for a transfer of knowledge from accurate but resource-heavy teacher models to lightweight student models, such that the latter can achieve competitive or even superior performance.

Knowledge distillation was popularized by Hinton et al. [133], where a student network is trained by mimicking the prediction outputs of the teacher network. Here, teacher knowledge takes the form of probabilities produced by the final temperature-controlled softmax, also referred to as soft targets. Using a transfer set (usually the original training set for the teacher model), the student model is learned by minimizing a weighted average of two objective functions: a distillation loss which measures the difference between the student's outputs with the soft targets, and a cross-entropy loss with the ground truth labels. Due to its simplicity and effectiveness, using soft targets for KD remains to be popular for guiding the student model. A number of extension works have investigated other effective forms of teacher knowledge, such as the activations, neurons, and features of intermediate layers [134], as well as their relationships.

Knowledge distillation is predominantly conducted *offline*, which means that the teacher model is assumed to have completed training (i.e. pre-trained) prior to the knowledge transfer process; This is different from *online* distribution frameworks where the parameters of the teacher model can still be updated during KD. Offline knowledge distillation offers the advantage of simplifying the implementation of KD, since the teacher knowledge may be extracted in the form of logits vectors and stored in a cache that is easily accessible during student model training. We refer readers to [135] which provides a comprehensive survey summarizing developments of KD methods, including recent works which explored other training schemes (e.g. online distillation where the teacher network is also updated during KD).

2.2.2.2 Cross-Modal Knowledge Distillation

Cross-Modal Knowledge Distillation is a subset of KD methods dedicated to transferring knowledge between teacher and student models which operate in different modalities. A motivating scenario for cross-modal knowledge distillation typically fulfill the following conditions:

1. The goal is to obtain a model that conducts a target task in the target modality B;
2. Very limited labeled data exists in target modality B (relative to modality A) for training a model for the target task;
3. A pre-trained teacher model (in modality A) is readily available; The task on which the teacher model was pre-trained is closely related to the target task;
4. A large cross-modal paired dataset exists, providing paired examples from modality A and modality B to facilitate the knowledge distillation process; This dataset need not be labeled with annotations for the target task.

Early examples of cross-modal knowledge distillation are rooted in the transfer of knowledge from RGB images to less common vision-based modalities, such as depth and optical flow [136, 137]. For instance, Gupta *et al.* are able to improve the state-of-the-art depth-based object detection performance [136], by exploiting existing paired RGB-D data to distill knowledge from teacher models pre-trained on ImageNet [138]. Following the success of these early works, many works investigated cross-modal knowledge distillation between vision and audio modalities [139–141]. They build upon the realization that videos provide a natural synchronization for vision and sound data, and thus large-scale paired cross-modal data can be obtained from online video repositories.

We group existing works into two main approaches by the way they incorporate source modality knowledge into student model training, and summarize them as follows:

Using Cross-modal Knowledge for Pre-training This is a two-stage approach that starts with a cross-modal pre-training step followed by a fine-tuning step. The pre-training stage conducts knowledge distillation, where the feature maps of the teacher model are used as targets for a student model in a new modality. We note that the cross-modal pre-training step in [136] can be interpreted as a special case of unsupervised pre-training, which introduces a useful prior to the later supervised fine-tuning training procedure [142].

If no annotated target modality data is available at all, the fine-tuning stage following cross-modal pre-training may be skipped entirely. Gupta *et al.* validated this idea by feeding the student-learned features into the classification head of the teacher model to generate predictions from target-modality inputs [136]. Alternatively, target-modality predictions can be produced by directly transferring response-based teacher knowledge (e.g. logits) in the knowledge distillation step [143, 144]. While this setup is well-suited for tasks where the annotations for the target modalities are almost impossible to obtain (e.g. human pose annotations using radio signals [143]), it is only limited to scenarios where the classification tasks in the source and target modality share the same label space.

Using Cross-modal Knowledge as Side Information Modality hallucination, introduced by Hoffman *et al.* [137], instead treats cross-modal knowledge as side information only available at training time. This formulation can be viewed as a *generalized distillation* framework [145], which unifies the teacher-student approach from knowledge distillation and additional knowledge provided by a privileged information source. In [137], they learn an additional mapping that hallucinates source-modality (depth) features from the target modality (RGB), by using an annotated paired dataset; At testing time, the resulting model simultaneously uses cross-modal representations and features specific of the target domain. They show that this approach produces richer RGB representations which lead to better test-time performance. This approach is particularly useful for situations where multi-modal data are only available for model training but not testing, and has been explored for wide-ranging applications such as pedestrian detection [146], hand articulations [147], face recognition [148], audio classification [149], thermal-inertial odometry [150], and video-based HAR [151].

Most works in cross-modal knowledge transfer have been successfully conducted using *paired* datasets. Paired datasets act as a bridge that transfers information between modalities via pairwise sample registration, though they can be difficult to obtain in many applications beyond vision and sound. Few works have examined *unpaired* training, amongst them, [152, 153] took a cycle-consistent adversarial training approach inspired by CycleGANs [154], while [155] proposed aligning the source and target probability distributions across classes.

2.2.3 Zero-Shot Learning

The machine learning methods that we have so far discussed focus on classifying instances whose classes have been pre-defined and seen during model development. However, in many practical applications of these models, for instance in activity recognition, these models will encounter instances whose classes had not been seen previously.

The paradigm of Zero-Shot Learning (ZSL) offers a powerful solution to address this challenging setting, in which the classes covered by training instances and the classes that we aim to classify at inference time are disjoint [156].

Problem Formulation We denote the set of seen classes as $\mathcal{S} = \{c_i^s\}_{i=1}^{N_s}$ and the set of unseen classes as $\mathcal{U} = \{c_i^u\}_{i=1}^{N_u}$. During model development, we have access to a training dataset $D_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{S}$, where N is the number of training instances, $\mathcal{X} \in \mathbb{R}^d$ is the input feature space, and all instances belong to the *seen* classes $\mathcal{S}$. We denote a testing dataset, $D_{test} = \{\mathbf{x}_i\}_{i=1}^{N_{test}} \in \mathcal{X}$, and these test instances belong to the *unseen* classes $\mathcal{U}$. Thus, the goal of zero-shot learning can be described as follows: Given training instances D_{train} belonging to $\mathcal{S}$, learn a classifier that can classify testing instances D_{test} belonging to $\mathcal{U}$.

Auxiliary Information The key to solving the zero-shot learning problem, where no labeled instances belonging to the unseen classes are available, is to make use of *auxiliary information*. The auxiliary information can be seen as a bridge between the unseen classes and the training instances belonging to the seen classes. Auxiliary information used by existing ZSL methods is usually in the form of a *semantic space* [156], which contains semantic information for both the seen and unseen classes by way of *class prototypes*. In the semantic space, each class has a corresponding real-valued vector representation, referred to as the prototype of this class. We denote the semantic space as $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{N_s+N_u}$, where $\mathbf{p}_k \in \mathbb{R}^F$ is the prototype for class c_k.

In zero-shot learning, the recognition method and the auxiliary information both play an important role. These are summarized as follows:

ZSL Recognition Methods Methods that achieve ZSL can be seen to exemplify two forms of knowledge transfer: (i) transferring knowledge from seen to unseen classes, (ii) transferring auxiliary knowledge from the semantic space to the feature space. A large group of methods that fulfill zero-shot learning adopt projection methods [157–160]. The key idea behind projection methods is to learn projection function(s) such that both the

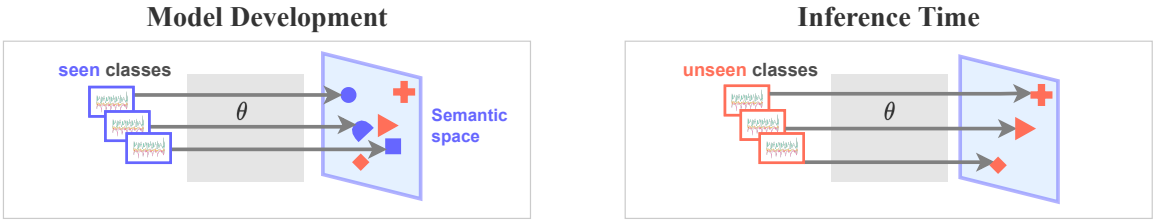


Figure 2.1: An illustration of projection-based methods for zero-shot learning. During model development, training instances belonging to the seen classes are used to learn the projection function θ, which projects instances from the feature space $\mathcal{X}$ to prototypes in the semantic space $\mathcal{P}$. At inference time, θ is used to project testing instances to the semantic space, where classification is performed using nearest neighbor classifiers or their variants.

instances from the feature space and the prototypes from the semantic space are cast into a target common space, from which labeled instances for unseen classes can be obtained. Illustrated in Figure 2.1 is one such setup, which uses the semantic space as the common space; Here, the key is to learn projection function θ such that instances in the feature space $\mathcal{X}$ are projected into the semantic space $\mathcal{P}$ as follows:

$$\mathcal{X} \rightarrow \mathcal{P} : \mathbf{z}_i = \theta(\mathbf{x}_i) \tag{2.4}$$

During model development, the projection function θ is learned using the training instances $\mathcal{D}_{train}$. At model inference time, testing instances from $\mathcal{D}_{test}$ are projected into the semantic space $\mathcal{P}$, and the projected instances can be classified to the nearest unseen class prototype via nearest neighbor classification. The use of nearest neighbor classification, a form of lazy learning method, is due to the limited number of labeled instances in the common space (the prototype is the only labeled instance of an unseen class). We refer interested readers to a comprehensive survey of ZSL methods carried by Wang *et al.* in [156], which further characterized existing ZSL methods into (i) classifier-based methods, which directly learn a classifier for the unseen classes; and (ii) instance-based methods, which seek to obtain labeled instances belonging to the unseen classes which can then be used for classification.

Semantic Space Construction The semantic information supplied by semantic spaces is pivotal to zero-shot learning. Semantic spaces used in existing works can be broadly summarized into hand-crafted or learned semantic spaces [156]. A popular example of hand-crafted semantic spaces are attribute spaces [161, 162], where a list of terms describing the properties of the classes are treated as attributes and used to form multi-hot vectors. Defining these attribute spaces requires domain-specific knowledge and is subjective and labor-intensive, especially when a large number of classes are involved in the recognition task. Alternatively, learned semantic spaces, especially those consisting of word vectors or text-based variants, are widely used. The idea behind learned semantic spaces is to derive a

vector representation of each class through auxiliary information that can be retrieved with ease, for example by embedding the class label [163], text description [164], or Internet-sourced image results [165] of the class.

2.3 Summary

In this chapter, we introduce the background to activity recognition from wearable inertial sensors, discussing the key issues surrounding inertial data collection and examining the steps involved to build activity recognition models. In the second half of this chapter, we describe three manifestations of cross-modal learning that will be investigated in this thesis: data generation, knowledge transfer, and zero-shot learning. These cross-modal learning methods have so far seen limited applications in inertial activity recognition, and they will require novel adaptations to be made to adequately overcome the unique challenges presented by this domain. In the following chapters, we build up on each of these ideas individually and present novel techniques that seek to further the state-of-the-art in Inertial HAR.

Chapter 3

Generating Virtual Inertial Data from Videos

3.1 Introduction

The collection of large-scale, labeled datasets has so far been limited in inertial human activity recognition. Yet access to such datasets is crucial to the promise of the inertial HAR problem – enabling the large-scale analysis of continuously collected behavioral data. Without models that can recognize a wide range of activities, a fine-grained and holistic behavioral analysis is precluded since one can only quantify human behaviours through the lens of a small group of, often coarsely defined, activity labels. Further, a fast feedback mechanism, prior to the staging of expensive recruitment-based data collection, may be enabled through easy access to a large source of activity data for *virtual* experimentation, in similar spirit to the widespread use of *in silico* experimentation before conducting expensive *in situ* studies in life sciences [166]. For instance, knowing in advance which of the hundreds of activities present in Kinetics [24–26] can an inertial HAR model reliably classify will be a useful signal to inform the design of in-person data collection studies, from what activities one should perform, to how fine-grain class labels should be defined.

In this chapter, we begin our exploration of cross-modal learning in the direction of virtual data generation – with the aim of lowering data collection and annotation costs in inertial HAR. Our goal is to develop a framework that can potentially alleviate the sparse data problem in inertial HAR. We aim at harvesting existing video data from large-scale repositories, such as YouTube, and automatically generate data for virtual, body-worn inertial sensors (IMUs) that will then be used for deriving human activity recognition systems that

can be used in real-world settings. The overarching idea is appealing due to the sheer size of common video repositories and the availability of labels in the form of video titles and descriptions. Having access to such data repositories opens up possibilities for more robust and potentially more complex activity recognition models that can be employed in entirely new application scenarios, which so far could not have been targeted due to the limited robustness of learned models.

The challenges for making these vast amounts of existing data usable for inertial activity recognition are manyfold, though: *i)* the datasets need to be curated and filtered towards the actual activities of interest; *ii)* even though video data capture the same information about activities in principle, sophisticated pre-processing is required to match the source and target sensing domains; *iii)* the opportunistic use of activity videos requires adaptations to account for contextual factors such as multiple scene changes, rapid camera orientation changes (landscape/portrait), the scale of the performer in the far sight, or multiple background people not involved in the activity; and *iv)* new forms of features and activity recognition models will need to be designed to overcome the shortcomings of learning from video-sourced motion information for eventual IMU-based inference.

In this chapter, we present a method that allows us to effectively use video data for training inertial activity recognizers, and as such demonstrates the first step towards larger-scale, and more complex deployment scenarios than what is considered the state-of-the-art in the field. Our approach extracts motion information from arbitrary human activity videos, and is thereby not limited to specific scenes or viewpoints. We have developed **IMUTube**, an automated processing pipeline that: *i)* applies standard pose tracking and 3D scene understanding techniques to estimate full 3D human motion from a video segment that captures a target activity; *ii)* translates the visual tracking information into virtual motion sensors (IMU) that are placed on dedicated body positions; *iii)* adapts the virtual IMU data towards the target domain through distribution matching; and *iv)* derives activity recognizers from the generated virtual sensor data, potentially enriched with small amounts of real sensor data. Our pipeline integrates a number of off-the-shelf computer vision and graphics techniques, so that IMUTube is fully automated and thus directly applicable to a rich variety of existing videos. One notable limitation of our current prototype is that it still requires human curation of videos to select appropriate activity content. However, with advances in computer vision, the potential of our approach can be further increased towards complete automation.

The work presented in this chapter is a first step towards the greater vision of automatically

deriving robust activity recognition systems for body-worn sensing systems. The key idea is to opportunistically utilize as much existing data and information as possible thereby not being limited to the particular target sensing modalities. We present the overall approach and relevant technical details and explore the potential of the approach in practical recognition scenarios. Through a series of experiments on three benchmark datasets—RealWorld [50], PAMAP2 [167], and Opportunity [40]—we demonstrate the effectiveness of our approach. We discuss the overall potential of models trained purely on virtual sensor data, which in certain cases can even reach recognition accuracies that are comparable to models that are trained only on actual sensor data. Moreover, we show that when adding only small portions of real sensor data during model training we are even able to outperform those models that were trained on real sensor data alone. As such, our experiments show the potential of the proposed approach, a paradigm shift for deriving inertial human activity recognition systems.

3.2 IMUTube

The key idea of our work is to replace the conventional data collection procedure that is typically employed for the development of wearable sensor-based human activity recognition (HAR) systems. Our approach aims at making existing, large-scale video repositories accessible for the HAR domain, leading to training datasets of sensor data, such as IMUs, that are potentially multiple orders of magnitude larger than what is standard today. With such a massively increased volume of *real* movement data—in contrast to simulated or generated samples, that often do not exhibit the required quality nor variability—it will become possible to develop substantially more complex and more robust activity recognition systems with a potentially much broader scope than the state-of-the-art in the field. In what follows, we first give an overview of the general approach before we provide the technical details of our procedure that converts videos into virtual IMU data.

3.2.1 Overview

Figure 3.1 gives on overview of our framework for deriving wearable sensor-based human activity recognition systems. The top left part (”conventional”) summarizes the currently

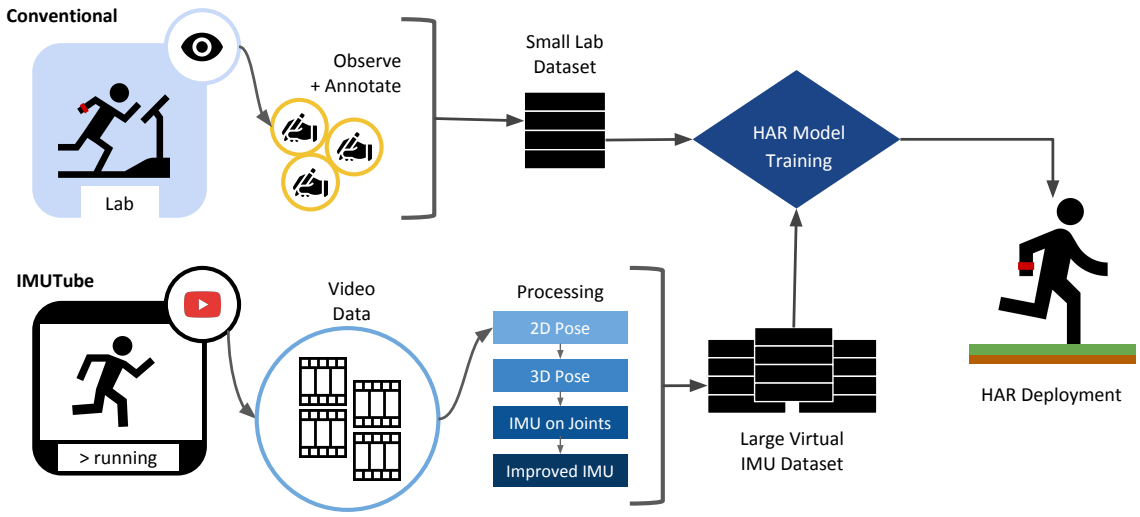


Figure 3.1: The proposed IMUTube system replaces the conventional data recording and annotation protocol (upper left) for developing sensor-based human activity recognition (HAR) systems (upper right). We utilize existing, large-scale video repositories from which we generate virtual IMU data that are then used for training the HAR system (bottom part).

predominant protocol. Study participants are recruited and invited for data collection in a laboratory environment. There they wear sensing platforms, such as a wrist-worn IMU, and engage in the activities of interest, typically in front of a camera. Human annotators provide ground truth labeling either directly, i.e., while the activities are performed, or based on the video footage from the recording session. This procedure is very labor-intensive and often error-prone, and, as such, labeled datasets of only a limited size can typically be recorded with reasonable efforts.

In contrast, our approach aims at utilizing existing, large-scale repositories of videos that capture activities of interest (bottom left part of Figure 3.1 labeled "IMUTube"). With the explosive growth of social media platforms, a virtually unlimited supply of labeled video is available online that we aim to utilize for training sensor-based HAR systems. In our envisioned application, a query for a specific activity delivers a (large) set of videos that seemingly capture the target activity. These results (currently) need to be curated in order to eliminate obvious outliers etc. such that the videos are actually relevant to the task (see discussion in Section 3.4). Our processing pipeline then converts the video data into usable virtual sensor (IMU) data. The procedure is based on a computer vision pipeline that first extracts 2D pose information, which is then lifted to 3D. Through tracking individual joints of the extracted 3D poses, we are then able to generate sensor data, such as tri-axial acceleration values, at many locations on the body. These values are then post-processed to match the target application domain.

Our work aims at replacing the data collection phase of HAR development. It is universal as it does not impose constraints on model training (top center in Figure 3.1) nor deployment (right part of Figure 3.1). In what follows, we describe the technical details of our processing pipeline that make videos usable for training inertial activity recognition systems. This description assumes direct access to a video that captures a target activity, i.e., here we do not focus on the logistics and practicalities of querying video repositories and curating the search results.

3.2.2 Motion Estimation for 3D Joints

On-body movement sensors capture local 3D joint motion, and, as such, our processing pipeline aims at reproducing this information but from 2D video. As shown in Figure 3.2 we employ a two-step approach. First, we estimate 2D pose skeletons for potentially multiple people in a scene using a state-of-the-art pose extractor, namely the *OpenPose* model [168]. Then, each 2D pose is lifted to 3D by estimating the depth information that is missing in 2D videos. Without limiting the general applicability we assume here that all people in a scene are performing the same activity. Although fast and accurate, the *OpenPose* model estimates 2D poses of people on a frame-by-frame basis only, i.e., no tracking is included which requires post-processing to establish and maintain person correspondences across frames. In response, we apply the *SORT* tracking algorithm [169] to track each person across the video sequence. *SORT* utilizes bipartite graph matching with the edge weights as the intersection-over-union (IOU) distance between boundary boxes of people from consecutive frames. The boundary boxes are derived as tight boxes including the 2D keypoints for each person.

To increase the reliability of the 2D pose detection and tracking, we remove 2D poses where over half of the joints are missing, and also drop sequences that are shorter than one second. For each sequence of a tracked person, we also interpolate and smooth missing or noisy keypoints in each frame using a Kalman filter, as poses cannot be dramatically different between subsequent frames. Finally, each 2D pose sequence is lifted to 3D pose by employing the *VideoPose3D* model [170]. Capturing the inherent smooth transition of 2D poses across the frames encourages more natural 3D motion in the final estimated (lifted) 3D pose.

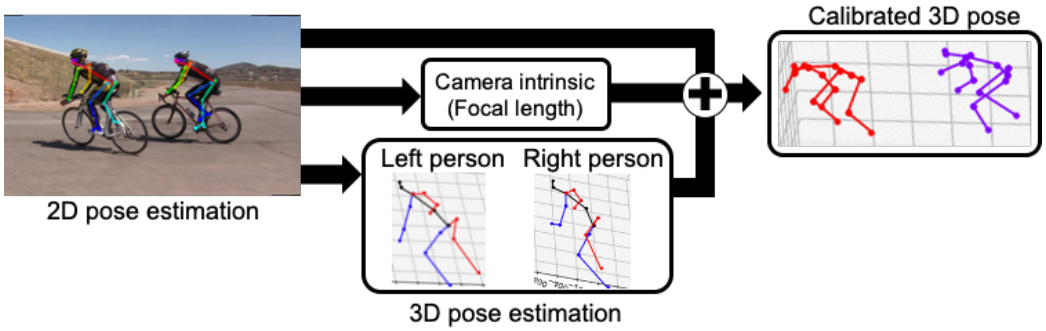


Figure 3.2: 3D joint orientation estimation and pose calibration. The multi-person 2D poses are estimated with *OpenPose* followed by lifting to 3D through *VideoPose3D*. The camera intrinsic parameters are estimated using the *DeepCalib* model. Jointly with the pose and camera related parameters, we calibrate the orientation and translation in the 3D scene for each frame.

3.2.3 Global Body Tracking in 3D

Inertial measurement units capture the acceleration from global body movement in 3D, and additionally local joint motion in 3D. Thus, we also have to extract global 3D scene information from the 2D video to track a person's movement in the whole scene. Typical 3D pose estimation models do not localize the global 3D position and orientation of the pose in the scene. Tracking the 3D position of a person in 2D video requires two pieces of information: *i)* 3D localization in each 2D frame; and *ii)* the camera viewpoint changes (ego-motion) between subsequent 3D scenes. We map the 3D pose of a frame to the corresponding position within the whole 3D scene in the video, compensating for the camera viewpoint of the frame. The sequence of the location and orientation of the 3D pose is the global body movement in the whole 3D space. For the virtual sensor, the global acceleration from the tracked sequence will be extracted along with local joint acceleration.

3D Pose Calibration

First, we estimate the 3D rotation and translation of the 3D pose within a frame, as shown in Figure 3.2. For each frame, we calibrate each 3D pose from a previously estimated 3D joint according to the perspective projection between corresponding 3D and 2D keypoints. The perspective projection can be estimated with the Perspective-n-Point (PnP) algorithm [171]. Additionally to 3D and 2D correspondences, the PnP algorithm requires the camera intrinsic parameters for the projection, which include focal length, image center, and lens distortion parameters [172, 173]. Since arbitrary online videos typically do not come with such metadata, the camera intrinsic parameters are estimated from the video

with the *DeepCalib* model [174]. The *DeepCalib* model is a frame-based model that considers a single image at a time so that the estimated intrinsic parameter for each frame slightly differs across the frame according to its scene structure. Hence, we assume that a given video clip sequence is recorded with a single camera, and aggregate the intrinsic parameter predictions by calculating the average from all frames:

$$c^{int} = \frac{1}{T}\sum_{t=1}^{T} c_t^{int} \tag{3.1}$$

where $c^{int} = [f, p, d]$ is the averaged camera intrinsic parameters from each frame, x_t at time t, predictions, $c_t^{int} = DeepCalib(x_t)$. $f = [f_x, f_y]$ is the focal length and $p = [p_x, p_y]$ is the optical center for the x and y axis, and d denotes the lense distortion. Once the camera intrinsic parameter is calculated, the PnP algorithm regresses global pose rotation and translation by minimizing the following objective function:

$$\begin{gathered}\{R^{calib}, T^{calib}\} = \arg\min_{R,T} \sum_{i=1}^{N} \|p_2^i - \frac{1}{s} c^{int}(Rp_3^i + T)\| \\ \text{subject to } R^T R = I_3, det(R) = 1\end{gathered} \tag{3.2}$$

where $p_2 \in \mathbb{R}^2$ and $p_3 \in \mathbb{R}^3$ are corresponding 2D and 3D keypoints. $R^{calib} \in \mathbb{R}^{3\times3}$ is the extrinsic rotation matrix, $T^{calib} \in \mathbb{R}^3$ is the extrinsic translation vector, and $s \in \mathbb{R}$ denotes the scaling factor [175, 176]. For the temporally smooth rotation and translation of a 3D pose across frames, we initialize the extrinsic parameter, R, and T, with the result from the previous frame. The 3D pose for each person, $p_3 \in \mathbb{R}^{3\times N}$, at each frame is calibrated (or localized) with the estimated corresponding extrinsic parameter.

$$p_3^{calib} = R^{calib} p_3 + T^{calib} \tag{3.3}$$

From the calibrated 3D poses, $p_3^{calib} \in \mathbb{R}^{3\times N}$, we remove people considered as the background. For example, in a rope jumping competition scene, a set of people may rope jump while others are sitting and watching. Depending on the scene, not all people captured may partake in an activity (e.g., bystanders). To effectively collect 3D pose and motion that belongs to a target activity, we thus remove those people in the (estimated) background. We first calculate the pose variation across the frames as the summation of the variance of each joint location across time. Subsequently, we only keep those people with the pose variation larger than the median of all people.

Estimation of Camera Ego-motion

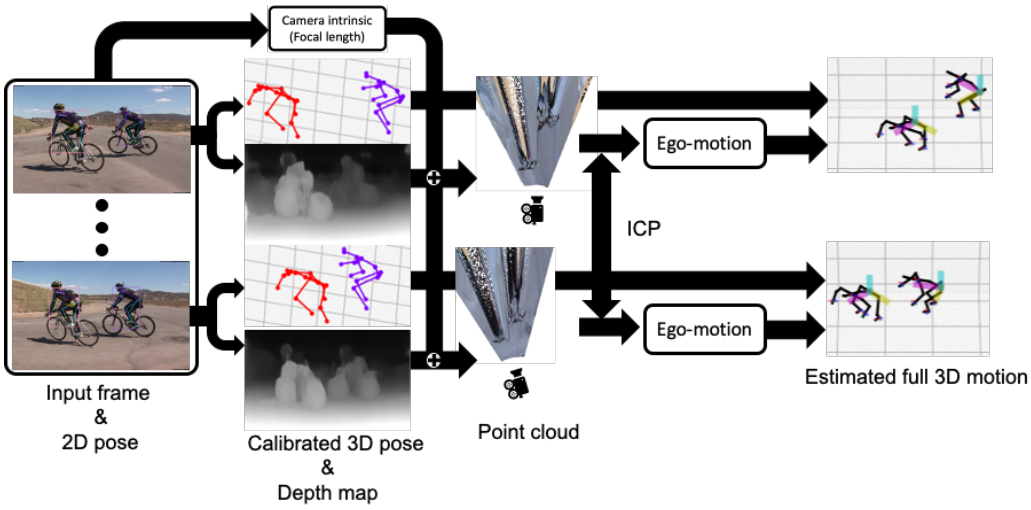


Figure 3.3: 3D pose and motion tracking with compensation of the camera motion. The camera motion is estimated through the iterative closest point (ICP) algorithm between subsequent point clouds. Then, calibrated 3D poses per frame are mapped to the location in the entire 3D scene, compensating for camera motion. The calibrated 3D poses from both frames are initially centered in the 3D world origin as the camera follows the cyclists. After incorporating ego-motion information, we can see that two cyclists are moving from right to left, moving closer to each other as in the video (most right figure).

In an arbitrary video, the camera can move around the scene freely. However, the pipeline should not confuse the camera motion with human motion. For example, a person who does not move (much) may appear at a different location in subsequent frames due to the camera movement, which is misleading for our purpose. Also, a moving person can always appear in the center of the frame, and thus erroneously appear static, if the camera follows that person and therefore the movements are effectively compensated for in the video. In these two cases, the virtual sensor should capture no motion (static), or the global body acceleration only, respectively, independently from camera motion. Hence, before generating the virtual sensor data, the 3D poses, which were previously calibrated per frame, need to be corrected for camera ego-motion, i.e., potential viewpoint changes, across the frames.

Camera ego-motion estimation from one viewpoint to another requires 3D point clouds of both scenes [177–179]. Deriving a 3D point cloud of a scene requires two pieces of information: *i)* the depth map; and *ii)* camera intrinsic parameters. For camera intrinsic parameters, we reuse the parameters previously estimated. The depth map is the distances of pixels in the 2D scene from a given camera center, which we estimate with the *DepthWild* model [180] for each frame. Once we have obtained the depth map and the camera intrinsic parameters, we can geometrically inverse the mapping of each pixel in the image to the

3D point cloud of the original 3D scene. With basic trigonometry, the point cloud can be derived from the depth map using the previously estimated camera intrinsic parameter, $c^{int} = [f_x, f_y, p_x, p_y, d]$. For a depth value Z at image position (x, y), the point cloud value, $[X, Y, Z]$, is:

$$[X, Y, Z] = \Big[\frac{(x - p_x) \cdot Z}{f_x}, \frac{(y - p_y) \cdot Z}{f_y}, Z\Big] \tag{3.4}$$

Once point clouds are calculated across frames, we can derive the camera ego-motion (rotation and translation) parameters between two consecutive frame point clouds. A popular method for registering groups of point clouds is the Iterative Closest Points (ICP) algorithm [177–179]. Fixing a point cloud as a reference, ICP iteratively finds the closest point pairs between two point clouds and estimates rotation and translation for the other point cloud that minimizes the positional error between matched points [178]. Since we extract color point clouds from video frames, we adopted Park *et al.*'s variant of the ICP algorithm [181], which considers color matching between matched points in addition to the surface normals to enhance color consistency after registration. More specifically, we utilize background point clouds instead of the entire point cloud from a scene because the observational changes for the stationary background objects in the scene are more relevant to the camera movement. We consider humans in the scene as foreground objects, and remove points that belong to human bounding boxes determined from 2D pose detection. The reason for this step is that we noticed that including foreground objects, such as humans, leads to the ICP algorithm confusing movements of moving objects, i.e., the humans, and of the camera. With the background point clouds, we apply the color ICP algorithm between point clouds at time $t-1$ and t, q_{t-1} and q_t respectively. As such, we iteratively solve:

$$\begin{aligned}\{R_t^{ego}, T_t^{ego}\} = \arg\min_{R,T} \sum_{(q_{t-1}, q_t) \in \mathcal{K}} \Bigg((1-\delta) \Big\| \mathcal{C}_{q_{t-1}}(f(Rq_t + T)) - \mathcal{C}(q_{t-1}) \Big\| \\ + \delta \Big\| (Rq_t + T - q_{t-1}) \cdot n_{q_{t-1}} \Big\| \Bigg)\end{aligned} \tag{3.5}$$

where $\mathcal{C}(q)$ is the color of point q, n_q is the normal of point q. $\mathcal{K}$ is the correspondence set between q_{t-1} and q_t, and $R_t^{ego} \in \mathbb{R}^{3\times3}$ and $T_T^{ego} \in \mathbb{R}^3$ are fitted rotation and translation vectors in the current iteration. $\delta \in [0, 1]$ is the weight parameter that balances the emphasis given to positional or color matches.

The estimated sequence of translation and rotation of a point cloud represents the resulting ego-motion of the camera. As the last step, we integrate the calibrated 3D pose and ego-

motion across the video to fully track 3D human motion. Previously calibrated 3D pose sequences, p_3^{calib}, are rotated and translated according to their ego-motion at frame t:

$$p_{3_t}^{track} = R_t^{ego} p_{3_t}^{calib} + T_t^{ego} \tag{3.6}$$

where $p_3^{track} \in \mathbb{R}^{T \times N \times 3}$ is the resulting 3D human pose and motion tracked in the scene for the video, T is the number of frames, and N is the number of joint keypoints. The overall process of compensating for camera ego-motion is illustrated in Figure 3.3.

3.2.4 Generating Virtual Sensor Data

Once the full 3D motion information has been extracted for each person in a video, we can extract virtual IMU sensor streams from specific body locations. The estimated 3D motion only tracks the locations of joint keypoints, i.e., those dedicated joints that are part of the 3D skeleton as it has been determined by the pose estimation process. However, in order to track how a virtual IMU sensor that is attached to such joints rotates while the person is moving, we also need to track the orientation change of that local joint. This tracking needs to be done from the perspective of the body coordinates. The local joint orientation changes can be calculated through forward kinematics based from the hip, i.e., the body center, to each joint. We utilize state-of-the-art 3D animation software – *Blender* [182], to estimate and track these orientation changes. Using the orientation derived from forward kinematics, the acceleration of joint movements in the world coordinate system is then transformed into the local sensor coordinate system. The angular velocity of the virtual sensor (i.e., a gyroscope) is calculated by tracking orientation changes.

We employ our video processing pipeline on raw 2D videos that can readily be retrieved by, for example, querying public repositories such as YouTube, and combined with subsequent curation (the latter is not within the focus of this Chapter). The pipeline produces virtual IMU, for example, tri-axial accelerometer data. This data effectively captures the recorded activities, yet the characteristics of the generated sensor data will still differ from real IMU data, for instance it will lack any MEMS noise. In order to compensate for this mismatch, we employ the *IMUSim* [122] model to extract realistic sensor behavior for each on-body location. *IMUSim* estimates sensor output considering mechanical and electronic components in the device, as well as the changes of a simulated magnetic field in the environment. As such, this post-processing step leads to more realistic IMU data [123, 183, 184].

3.2.5 Distribution Mapping for Virtual Sensor Data

As the last step before using the virtual IMU dataset for HAR model training, we define a calibration operation to account for any potential mismatch between the source (virtual) and target (real) domains. We employ a distribution mapping technique to fix such mismatch, where we transfer the distribution of the virtual sensor to that of the target sensor. For computational efficiency, the rank transformation approach [185] is utilized:

$$x_r = G^{-1}(F(X \leq x_v)) \tag{3.7}$$

where, $G(X \leq x_r) = \int_{-\inf}^{x_r} g(x)dx$ and $F(X \leq x_v) = \int_{-\infty}^{x_v} f(x)dx$ are the cumulative density functions for real, x_r, and virtual, x_v, sensor samples, respectively. In our experiments (Section 3.3.2.2), we show that only a few seconds to minutes of real sensor data is sufficient to calibrate the virtual sensor effectively for successful activity recognition.

3.3 Experiments

This section contains an extensive empirical evaluation of the proposed IMUTube pipeline, organized into three main experimental sections. First, we begin with the first set of experiments that examine HAR model performance when trained using real IMU data and/or virtual IMU data generated with IMUTube under varying conditions. Next, we delve into the empirical observations to provide readers with a better understanding of the virtual IMU data and its limitations. Finally, we explore an alternative training setup using transfer learning and provide its empirical evaluation.

3.3.1 Establishing Activity Recognition Feasibility

In this section, we seek to empirically examine the viability of using IMUTube to produce virtual IMU data useful for HAR. The core question we aim to answer is, *under what conditions can the virtual IMU data extracted using IMUTube be of value to building inertial HAR models?*

To answer this question, we present three experiments under a varying set of constraints on the employed video and IMU data; the datasets used and motivations for doing so in each

experiment will be described in detail. Our general experimental framework compares the performance of HAR models under three settings: **R2R**, where real IMU data are used to train and evaluate the model; **V2R**, where Virtual IMU data are used to train the model, which is then evaluated on real IMU data; and **Mix2R**, where a mixture of virtual and real IMU data are used to train the model, which is then evaluated on real IMU data.

Throughout this chapter, our primary evaluations are performed with the Random Forest classifier using 15-component ECDF features [78], trained using a leave-one-subject-out scheme. We calibrate hyperparameters on the validation subjects by varying the number of trees from 3 to 50 and the minimum number of samples in leaf node from 1 to 50. We report the mean F1-score of the test subjects computed after the completion of all folds.

Separate from this, we also train ConvLSTM [86] on a hold-out evaluation scheme, where data from random subject(s) are selected as the validation and test set. ConvLSTM is trained on raw data for a maximum of 100 epochs with an Adam optimizer [186] and early stopping on the validation set with a patience of ten epochs; the learning rate is searched from 10^{-6} to 10^{-3}, and weight decay from 10^{-4} to 10^{-3} via grid search. We further regularize model training by employing augmentation techniques from [187] with a probability of application set at either 0 and 0.5 depending on validation set result. We average over 3 runs initiated with different random seeds and report the mean F1-score.

By supplementing our primary random forest results with ConvLSTM, we aim to examine IMUTube's feasibility in both the classical machine learning and deep learning regime, and show that our approach is agnostic to the choice of specific learning algorithm. In both cases, we report the highest test F1-score achieved using any amount of training data, along with the Wilson score interval with 95% confidence. For V2R and Mix2R, we also report the absolute difference in F1-scores achieved by them and the baseline R2R setting.

3.3.1.1 Controlled Conditions

There are many potential sources of noise that may impact the activity recognition performance; therefore, in our first experiment we hold constant as many factors as possible between real and virtual IMU data to establish a fair comparison between them. We accomplish this by using the RealWorld dataset [50], an activity recognition dataset that contains not only IMU data but also provides videos of the subjects performing the activities. A summary of the Realworld dataset is provided in Section 2.1.1.2.

Dataset	Model	R2R	V2R	Mix2R
Realworld	RF	57.8 ± 0.3	56.8 ± 0.3 (**-1.0**)	64.4 ± 0.3 (**+6.6**)
	ConvLSTM	73.1 ± 0.7	54.7 ± 0.8 (**-18.4**)	77.9 ± 0.7 (**+4.8**)

Table 3.1: Recognition results on the Realworld dataset (8 classes) when training models from real IMU data (R2R), from virtual IMU data (V2R), and from a mixture of both (Mix2R). Wilson score confidence intervals are shown. For Random Forest models, V2R achieves 98% of the R2R F1-score, while a Mix2R setup surpasses R2R by 12%. Quoted in brackets is the difference in F1-scores between each model and its R2R baselines.

Method

The unconstrained settings under which the Realworld dataset was recorded present extra difficulty in extracting virtual IMU for the full duration of the activities; nonetheless we are able to extract 12 hours of virtual IMU data, this is compared to 20 hours of available real IMU data. As a preprocessing step, we remove the first ten seconds of each video and divide the remainder into two-minute chunks for the efficient running of IMUTube. Virtual IMU data are extracted from 7 body locations corresponding to where real sensors are placed in Realworld. We assume all IMU data to have a frequency of 30Hz and use sliding windows to generate training samples of duration 1 second and 50% overlap. The resulting real and virtual IMU dataset contains $221k$ and $86k$ windows, respectively.

Results

In Table 3.1, we see convincing evidence that human activity classifiers can learn from virtual IMU data alone. When learning from virtual IMU data alone (V2R), the 8-class model achieves an F1-score of 56.8, which is within 2% of that achieved by learning from real IMU data (R2R). This result is remarkable as the difference in recognition performance of R2R and V2R is small notwithstanding the change in data source and the introduction of noise while going through our pipeline.

Furthermore, when we use a mixture of virtual and real IMU data to train the model, it is even able to surpass R2R performance with a significant relative performance gain of 11%, reaching an F1-score of 64.4. This showcases an additional potential of IMUTube – we can build activity classifiers using both virtual and real IMU data to push recognition capabilities beyond that achieved by either.

Our ConvLSTM results (evaluated on a random subject) offers another perspective into modeling virtual IMU data when deep learning models are used. Learning from virtual

IMU data alone is seen to pose more challenges (V2R achieves 75% of R2R), which is possibly related to the setup of learning directly from raw data, as opposed to learning from processed features in the Random Forest case. As a consequence, ConvLSTM may be learning feature representations that are highly specific to the virtual IMU domain, which prevents immediate generalization to the target domain of real IMU data. This issue is diminished when using a mixture of virtual and real IMU data for training, as Mix2R even outperforms R2R by 6.6%. We presume that the improvement is related to the complementary benefits of both real and virtual data, as well as the feature learning capabilities of deep learning models, which learn better when more data is available.

This set of results provides promising signs for IMUTube – we can learn capable activity classifiers with virtual IMU data alone, despite having only so far considered relatively straightforward techniques in extracting and modeling the virtual IMU data. We delve into these concerns about the quality of virtual IMU data in Section 3.3.2 to provide a more complete view.

3.3.1.2 Common Activity Recognition Datasets

In Section 3.3.1.1, we saw promising results under a controlled setup on the Realworld dataset, which gathers video and IMU data for the same set of activities, performed by the same set of users. This, however, is a unique setup since most inertial HAR datasets do not provide video data. Thus, we now seek to relax these conditions, and establish the viability of IMUTube when the activities performed in the video data and the real IMU data do not completely align.

Imagine a scenario where we want to build a HAR classifier for "standing" vs. "sitting". Instead of collecting simultaneous video and real IMU data of people standing and sitting, we want to exploit existing videos of people standing and sitting, extract their virtual IMU data via IMUTube, and use them to train a classifier that operates on real IMU data. Such is the scenario we seek to recreate in this experiment. Specifically, we extract virtual IMU data using videos collected in the RealWorld dataset, and evaluate the trained models on real IMU data taken from two other HAR datasets, Opportunity [40] and PAMAP2 [167] (introduced in Section 2.1.1.2).

Method

Dataset	Model	R2R	V2R	Mix2R
Opportunity	RF	82.7 ± 0.3	77.6 ± 0.4 (**-5.1**)	88.2 ± 0.3 (**+5.5**)
	ConvLSTM	88.7 ± 0.7	78.8 ± 1.0 (**-9.9**)	88.4 ± 0.8 (**-0.3**)
PAMAP2 (8-cls.)	RF	70.3 ± 0.6	67.3 ± 0.6 (**-3.0**)	72.8 ± 0.5 (**+2.5**)
	ConvLSTM	70.0 ± 1.6	55.2 ± 1.8 (**-14.8**)	70.2 ± 1.6 (**+0.2**)

Table 3.2: Recognition results (mean F1-score) on Opportunity dataset (4 classes) and locomotion activities found in PAMAP2 (8 classes) when using different training data. For Random Forest models, V2R achieves 95% of R2R F1-scores on average, while Mix2R outperforms R2R by 5% on average.

Opportunity and PAMAP2 are used as they contain activity labels that roughly correspond to the 8 classes in Realworld. Here, we only consider activities in Opportunity and PAMAP2 which are overlapping with those in Realworld, i.e., 4 classes (*stand, walk, sit, lie*) in Opportunity, and 8 classes (*ascending stairs, descending stairs, rope jumping, lying, running, sitting, standing, walking*) in PAMAP2. We again segment the data using 1-second sliding windows with 50% overlap.

For Opportunity, we re-extracted virtual data from the Realworld videos in eleven body positions: left and right feet, left shin and thigh, hip, back, left and right arms, left and right forearms, resulting in $40k$ and $46k$ real and virtual IMU windows respectively. For ConvLSTM, we used random subject 3 for validation, 4 for testing, and the rest for training.

For PAMAP2, the activities are slightly different from those in Realworld so we equated the labels with the closest meaning (e.g., using *jumping* Realworld videos as the source for *rope jumping* virtual IMU in PAMAP2. The PAMAP2 dataset specifies that sensors were placed in three locations (dominant wrist, dominant ankle, chest), which gives rise to a total of four possible combinations for arm and chest location when we extract virtual IMU data from a single video (i.e., left-left, right-right, left-right, right-left). We took advantage of this ambiguity and extracted $4\times$ as much virtual IMU per video, resulting in $24k$ and $152k$ windows for real and virtual IMU respectively. For ConvLSTM, we followed the same setup as [188] and use subject 5 for validation, 6 for testing, and the rest for training.

Results

Table 3.2 shows the classification performance for R2R, V2R and Mix2R. We observe encouraging results, where learning from virtual IMU data alone can still recover high levels of R2R performance, despite data collection conditions not being held constant. From

the V2R results, we observe that Random Forest classifiers trained from virtual IMU data achieve 94% and 96% of R2R performance on Opportunity and PAMAP2 respectively. While this good performance might be related to the simplicity of the motions classified (mainly locomotive activities), we highlight that the conditions of data collection in Realworld and Opportunity are very different–subjects could be walking through the forest or city in Realworld, but all subjects perform activities inside a laboratory in Opportunity; Likewise for Realworld and PAMAP2–subjects could also be climbing down the streets of a city (a mixture of pavement and stairs) in Realworld whereas all subjects are climbing up the same building in PAMAP2. Thus, being able to utilize virtual data from one scenario and test it on another is not a trivial task. These results show that, across these two tasks, IMUTube-extracted virtual data provide salient features which are generalizable and robust across testing scenarios.

We again observe performance gains when training with a mixture of real and virtual IMU data, which exceeds R2R F1-scores by 5% on average. Not only does this observation solidify the argument that virtual and real IMU data can bring complementary benefits to activity recognition, but such performance gains are also a positive sign especially since the two types of data are collected under rather different circumstances. We argue that, adding virtual IMU data – in this case, virtual data generated from a related but different scenario – help expand the intra-class variety of motions seen by the classifier and as a result improve model generalization.

As before, we provide the performance by ConvLSTM on a random test subject as additional results in Table 3.2, where V2R recovers 84% of R2R F1-scores, while Mix2R and R2R scores are statistically comparable.

Overall, this set of results presents strong evidence supporting the usefulness of virtual IMU data, either used standalone or in combination with real IMU data for activity recognition. Beyond this, these results also imply an encouraging view that aligns well with our vision for IMUTube – that virtual data, even when collected under vastly different settings, can be useful in building capable or even better models for activity recognition.

3.3.1.3 Complex Activity Recognition

Encouraged by the results so far, we now try to apply IMUTube to activity recognition scenarios with even more challenging conditions and test its ability in building classifiers

Dataset	Model	R2R	V2R	Mix2R
PAMAP2	RF	72.3 ± 0.4	57.9 ± 0.5 (**-14.4**)	71.1 ± 0.4 (**-1.2**)
(11-cls.)	ConvLSTM	69.8 ± 1.3	53.3 ± 1.4 (**-16.6**)	71.0 ± 1.3 (**+1.2**)

Table 3.3: Recognition results (mean F1-score) on PAMAP2 (11-cls.) when using different training data. The Random Forest model trained with virtual IMU data including YouTube videos (for four complex activities) recovered 80% of R2R model performance.

for complex activities. Our ultimate vision for IMUTube is to extract virtual data from any video, especially those freely available in large online repositories such as YouTube. To test the feasibility of doing so, we first need to curate a dataset with activity videos originating from the web. In the following, we attempt to source these videos for complex activities present in PAMAP2, and train classifiers with the extracted virtual IMU data.

Method

We curated a dataset of virtual data covering four complex activities present in PAMAP2, namely *vacuum cleaning, ironing, rope jumping* and *cycling*. To efficiently locate such videos, we extract annotated video segments from activity video datasets in the computer vision domain, including ActivityNet [189], Kinetics700 [190], HMDB51 [191], MPI-IHPD [192], UCF101 [193], Charades [194], AVA [76], MSRdailyactivity3D [195], and NTU RGB-D [196]. The resulting video dataset consists of a mix of videos collected in experiment scenarios and in-the-wild (e.g., from YouTube). In total, we collected ~ 10 hours of virtual data from 7,255 videos. To extend our activity recognition task to as many classes in PAMAP2 as possible, we also reuse the other seven videos from Realworld (we do not use the *jumping* videos); this allows us to consider an 11-class activity recognition problem in PAMAP2. As mentioned in Section 3.3.1.2, we face an ambiguity in sensor location which lead us to extract $4\times$ virtual data per video. Using sliding windows of 1-second size and 50% overlap resulted in $38k$ real and $390k$ virtual IMU windows in total.

Results

Our results in Table 3.3 show that these difficult conditions (using in-the-wild videos, and learning complex activities) indeed present challenges to the use of virtual IMU data extracted by our current formulation of IMUTube.

With the Random Forest classifier, training from virtual IMU data alone achieves a 0.58 F1-score under V2R, which is 80% of that achieved with R2R (0.72 F1-score); This is a weaker

result compared to those achieved on previous datasets (where V2R achieved 96% of R2R on average). When real IMU data is added to virtual IMU data for training, the Random Forest model gains 23% and achieves a F1-score of 71.1 in Mix2R versus 57.9 in V2R, though Mix2R is still 2% worse than R2R performance. Our evaluation using ConvLSTM is also shown in Table 3.3, and their results align with those of Random Forest. We believe that the weaker performance is likely linked to the even more drastic difference in the data sources and activity label interpretations between the real and virtual IMU data. Another cause may be the quality of virtual IMU data that IMUTube is currently able to produce, as noises may be amplified by the complex activities introduced.

Through this set of experiments, we have shown the limitations of IMUTube when operating under very challenging conditions – in-the-wild videos and complex activity recognition. While it is still feasible to learn a reasonably capable classifier using the extracted virtual IMU data in this case, we have not yet observed significant gains provided by virtual IMU data. These observations prompt us to understand the sources of these limitations and explore ways to combat them in the upcoming sections. In particular, in Section 3.3.2.1 we will examine the fidelity of virtual IMU data, and provide directions to improve its quality in Section 3.4. We discuss issues of domain shift in Section 3.3.2.2. Moreover, to better utilize both real and virtual IMU data, we investigate the effect of mixing data in Section 3.3.2.3 and investigate more sophisticated methods beyond simple mixing in Section 3.3.3.

Overall, our results in this section suggest promising signs that IMUTube can already generate useful virtual IMU data for real-world instances of activity recognition. Demonstrated over a range of activity recognition tasks, virtual IMU data is seen to effectively capture motion information, such that classifiers can be trained from them alone and still perform well on real IMU data. In addition, mixing real and virtual IMU data for training is also shown to be a potential source of performance gain.

3.3.2 Understanding Virtual IMU Data

In the previous set of experiments (Section 3.3.1), we see that models trained on the virtual IMU data (V2R) performed reasonably well when tested on real IMU data across different scenarios. Amongst common non-complex activity recognition tasks, V2R models achieve up to 94 to 98% of the performance of their R2R counterparts, and even outperform them by 4 to 11% when trained alongside real IMU data (Mix2R). However, for the experiment on complex activity recognition using data from PAMAP2 (11-class), the V2R model

could not outperform R2R even when a larger virtual dataset extracted from multiple video sources was used.

Thus, in this section, we investigate potential sources of such performance gaps in detail. First, we analyze the extracted virtual IMU data by inspecting the sample-level similarity in IMU signals using synchronized sequences of real and virtual IMU data. Then, at a distribution level, we investigate the effects of domain shift, along with the impact of our distribution mapping technique (described in Section 3.2.5). Finally, we investigate the mixing of real and virtual IMU data for model training (Mix2R), which is seen to always give comparable, and often superior, performance relative to R2R in Section 3.3.1. With our analysis, we aim to provide key insights into the IMUTube pipeline and the use of virtual IMU data for human activity recognition. All experiments presented in this section are carried out using Random Forest in the leave-one-subject-out setting unless otherwise specified.

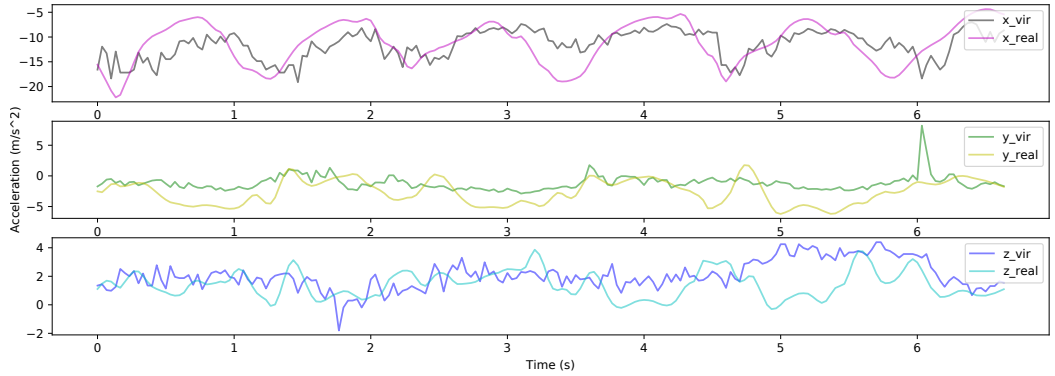
3.3.2.1 Comparing Virtual and Real IMU

We do not expect IMUTube to function flawlessly. Given the complexity of the process, the translation from video to virtual IMU data will naturally contain errors. Despite this, we observe promising results of competitive V2R performance in prior experiments. This seems to suggest that perfect sample-level realism in virtual IMU data is not necessary to train capable human activity classifiers. In the following, we compare virtual and real IMU samples to better understand the limits of IMUTube and argue that, the focus, during virtual IMU generation, should be placed on capturing salient features useful for activity recognition.

Sample-level Comparison

Sample-level comparison between the virtual and real IMU data requires a dataset with time-synchronized video and IMU data. Although the Realworld dataset provides both accelerometer and video data, these modalities are not time-synchronized. Therefore, in this experiment, we adopt the TotalCapture dataset [197] which contains time-synchronized (real) IMU data and video recordings (from which the virtual IMU data are extracted). As TotalCapture contains various scripted motions but not labels that distinguish activity types, this dataset is not usually used for activity recognition-related tasks and we have not evaluated on TotalCapture in Section 3.3.1.

(a) Left wrist



(b) Left ankle

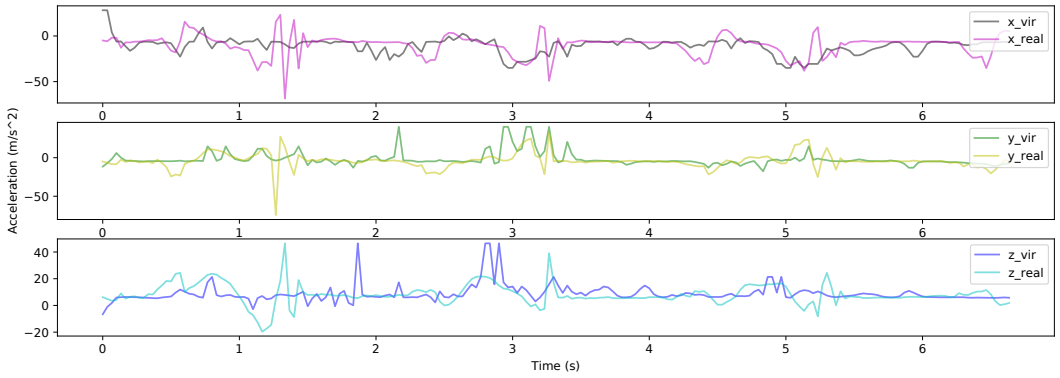


Figure 3.4: Comparison between virtual and real IMU on the TotalCapture dataset for a user who is performing the "walking" activity. The virtual IMU data is generated by passing a video clip (which is time-synchronous with the real IMU data) through IMUTube with distribution mapping applied. The aligned time series are depicted for sensors placed on the user's left wrist in (a) and left ankle in (b).

Figure 3.4 shows an example of the virtual and real IMU time-series data of a subject walking, with sensors placed on their wrist and ankle. Along the x-axis, virtual IMU readings are seen to reflect large movement changes also observed in the real IMU—one can almost see from the "wrist" time-series (see Figure 3.4(a)) that the person is walking with periodic hand movements. Along the z-axis, the virtual IMU is also seen to capture any spikes in acceleration reasonably well, albeit with a noticeable time lag in the "ankle" case. Virtual and real IMU data differ the most along the y-axis. We postulate that this is related to a dimensionality issue—we are trying to reconstruct 3-D information from a 2-D image time series. The y-axis here refers to the axis pointing perpendicular to the visual plane, which means any acceleration measured along this dimension cannot be easily deduced visually.

While generating realistic virtual IMU data is important, it is secondary to our main goal of producing virtual IMU data that captures useful information for HAR tasks. To achieve this, what is vital is the ability of the virtual IMU data to capture salient features of the activities that we need to recognize. We already see signs of this happening with the current IMUTube (e.g., the x-axis of the "arm" while walking in Figure 3.4). This also offers a possible explanation for the better V2R performance seen in predicting locomotion-style activities in Section 3.3.1. Perhaps IMUTube, in its current form, is best suited to capture information about simple motions (i.e., ones mostly characterized by movement in a 2D plane) of which there is still a wide variety, and to which existing HAR methods still struggle to generalize [198] (for additional qualitative observations see Section 3.4). To apply IMUTube to more complex activities, it may require improved techniques during virtual IMU data generation (also discussed further in Section 3.4).

3.3.2.2 Coping with Domain Shift

The last step of IMUTube performs a distribution mapping post-processing step between virtual and real IMU data (Section 3.2.5), in order to correct for the *domain shift* that exists between training and testing data. In the following, we pose two related questions and aim to justify the application of such a distribution mapping method through empirical analysis.

By answering these questions, we argue that: (1) such domain shift is not exclusive to video-generated virtual IMU data and in fact common whenever IMU data is taken from different datasets (or measured from different IMU sensors) during the training and testing phase, and (2) only limited amounts of real IMU data is needed in our adopted distribution mapping method.

Does IMUTube unnecessarily introduce domain shift?

Here, we seek to demonstrate that domain shift is not exclusive to virtual IMU data generated from videos through IMUTube, but in fact it is present whenever IMU data is taken from different datasets (or measured from different IMU sensors) which result in dissimilar data distributions between training and testing [61].

We quantify "domain shift" as the drop in F1 scores by a HAR model where a training dataset that is dissimilar to the testing dataset is used, when compared to a HAR model where the train and test set are different subsets of the same dataset. Our experiment is then to compare the recognition performance on the Opportunity dataset with models trained using data from sources other than Opportunity, with or without distribution mapping. Specifically, the train data can be *i)* real IMU data from PAMAP2, *ii)* real IMU data from Realworld, or *iii)* virtual IMU data from Realworld videos (Section 3.3.1.1). Without distribution mapping, we use all available data in the respective datasets that fall under the 4 Opportunity classes (stand, walk, sit, lie) for training, and testing using the entire Opportunity dataset. With distribution mapping, we follow a leave-one-subject-out evaluation scheme; In each fold, we train the model using data distribution-mapped with data only from the corresponding train subjects in Opportunity, and evaluate the remaining test subject.

In the "Before Distribution Mapping" column of Table 3.4, the effects of domain shift are demonstrated by a significant drop across all models trained from data not from the Opportunity dataset. When using training data not from Opportunity – despite having the same exact activity labels – even models trained with real IMU data (from PAMAP2 and Realworld) suffer a 66% drop in F1-scores. From this, it is clear that the domain shift issue is not exclusive to the distribution shift present between virtual and real IMU domains. The drop in performance is resolved when we perform distribution mapping (Section 3.2.5); we even observe that training from virtual data outperforms training using other real IMU datasets. This hints that virtual data might have greater value than real IMU data in developing general HAR models. This conclusion was also supported by the results seen when performing the same analysis on the PAMAP2 and Realworld datasets.

How much real IMU data is needed for distribution mapping?

To answer this question, we focus on the virtual and real IMU in the 4 datasets evaluated in Section 3.3.1, i.e., Realworld, Opportunity, PAMAP2 8-class and 11-class. We vary the

amount of real IMU data used for distribution mapping and evaluate at what point the virtual and real IMU data distribution become sufficiently similar. We report the similarity between each data distribution using the Frechet Inception Distance (FID), a metric commonly used in generative modeling to compare the real and generated datasets [199, 200]; lower FID scores indicate more similar data distributions. We also report the confidence interval as calculated by randomly sampling real IMU data with 10 different random seeds for distribution mapping.

Figure 3.5 shows how the virtual/real FID score varies with the quantity of real IMU data used for distribution mapping. In all four cases, an abrupt, significant drop in the FID score is seen with the use of under 100 seconds of real IMU samples. When using 10 minutes of real IMU data per class for distribution mapping, the FID scores are within 6% of the final FID score (when all real IMU is used for distribution mapping).

3.3.2.3 Varying the Mixture and Size of Training Data

We have seen that using a mixture of real and virtual IMU for training HAR models may bring about performance gains over a traditional R2R setup in Section 3.3.1. Here, we seek to understand how such performance gain may be affected by the size and composition of the training dataset. Again we consider the four datasets introduced in Section 3.3.1.

How does performance vary with increasing real IMU data?

Our first experiment compares the F1-scores achieved by models trained with a mixture of real and virtual IMU data (fixed at 1:1 ratio) as the amount of real training data is varied. We repeat this on all 4 datasets and plot the respective learning curves in Figure 3.6 to inspect if the performance gains by Mix2R over R2R are consistent.

For 3 out of 4 datasets, Mix2R outperforms R2R consistently by a clear margin at every inspected point of the learning curves. The greatest performance gain is observed throughout the Realworld learning curve, with an increase of at least 9% in F1-score by Mix2R over R2R. Similar trajectories are observed on Opportunity and PAMAP2 (8-class), with the most significant difference between Mix2R and R2R occurring when very limited training data are available (under 100 seconds per class).

On PAMAP2 (11-class), Mix2R outperforms R2R when there are only 10 samples per class, and both Mix2R and R2R curves plateau when there are more data available. Given

Source of Data (Train → Test)	**Before Distribution Mapping**	**After Distribution Mapping**
Virtual IMU data → Opportunity	29.5 ± 0.4 (**-53.2**)	77.6 ± 0.4 (**-5.1**)
PAMAP2 → Opportunity	27.7 ± 0.4 (**-55.0**)	69.3 ± 0.4 (**-13.4**)
Realworld → Opportunity	28.3 ± 0.4 (**-54.4**)	66.4 ± 0.4 (**-16.3**)
Opportunity → Opportunity	82.7 ± 0.3	-

Table 3.4: Recognition results (mean F1-score, Random Forest) on the 4-class activity recognition task from Opportunity when using training data from other data sources (top rows) and from Opportunity itself (last row, provided for reference). Without distribution mapping, there is a significant drop in performance when using training data collected under different circumstances than the test case, regardless of whether the IMU data is real (66%) or virtual (64%). This is resolved when distribution mapping is applied.

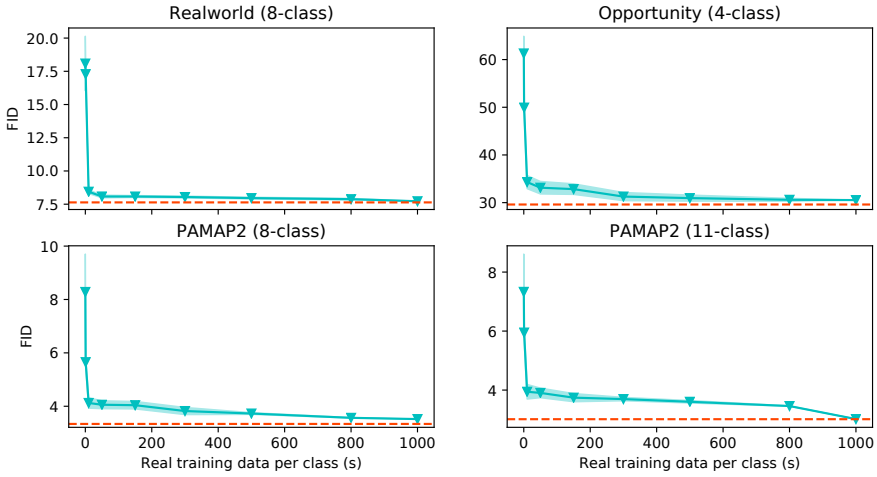


Figure 3.5: Frechet Inception Distance (FID) and confidence interval (shaded area) between virtual IMU and real IMU data distributions after performing distribution mapping of the virtual IMU data with increasing amounts of real IMU samples. The dotted horizontal line is the FID score obtained when the entire real IMU dataset is used for distribution mapping.

that PAMAP2 (11-class) is also the case where we predict complex activities under the most dissimilar settings, the plot highlights the difficulty of the classification task for both real and virtual IMU data.

Although it may appear that the learning performance saturates with a relatively small amount of data per class (600 seconds per class, for instance) – we highlight that this has been commonly observed in the literature for the HAR datasets we used (Opportunity, PAMAP2, e.g., [201–203]).

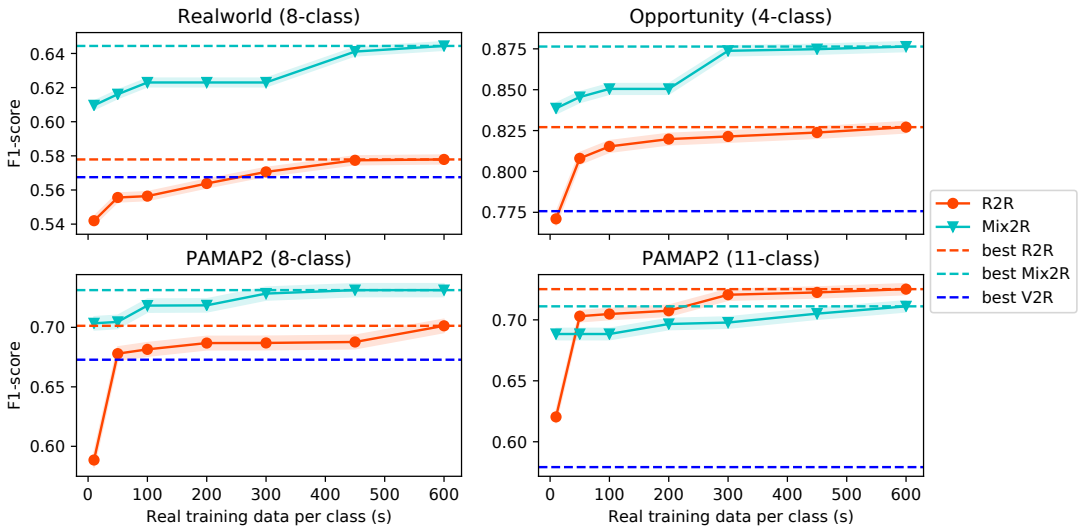


Figure 3.6: Mix2R and R2R performance of a random forest model on 4 different HAR tasks when different amounts of real data per class (in seconds) are used for training. The ratio of virtual data and real data is kept at 1:1 at all datapoints.

	Real:Virtual	Realworld	Opportunity	PAMAP2 (8-cls.)	PAMAP2 (11-cls.)
R2R	1:0	57.1 ± 0.3	82.1 ± 0.3	68.7 ± 0.6	72.1 ± 0.4
Mix2R	1:1	62.3 ± 0.3 (**+5.2**)	87.4 ± 0.3 (**+5.3**)	72.8 ± 0.5 (**+4.1**)	69.8 ± 0.5 (**-2.3**)
	1:2	61.5 ± 0.3 (**+4.4**)	86.4 ± 0.3 (**+4.3**)	70.1 ± 0.6 (**+1.3**)	70.5 ± 0.5 (**-1.6**)
	1:5	62.5 ± 0.2 (**+5.4**)	85.0 ± 0.3 (**+2.9**)	69.3 ± 0.6 (**+0.6**)	69.0 ± 0.5 (**-3.1**)
	1:10	60.6 ± 0.3 (**+3.5**)	84.0 ± 0.3 (**+1.9**)	67.9 ± 0.6 (**-0.8**)	68.2 ± 0.5 (**-3.9**)

Table 3.5: Recognition results (mean F1-score, Random Forest classifier) on all datasets. Different amounts of virtual IMU data are added to real IMU data, kept constant at 300 seconds per class.

How does performance vary with increasing virtual IMU data?

Our second experiment compares the F1-score achieved by models trained with a mixture of real and virtual IMU data, where the real IMU data is fixed at 300 seconds per class, but the real-to-virtual data ratio is varied from 1:1 to 1:10.

We evaluate the performance achieved when varying virtual and real IMU data mixtures, as presented in Table 3.5. When compared to the F1-scores of models only trained from real data, adding virtual data to training is seen to give a better or comparable performance at all considered ratios on Realworld, Opportunity, and PAMAP2 (8-class). On Realworld, the greatest gain is seen at 1:5, where the F1-score is increased by 9% over that at 1:0; At the ratio 1:1, both Opportunity and PAMAP2 (8-class) see improvements of 6%. The

performance however does not increase monotonically with the addition of more virtual data. This shows that the effect of mixing virtual and real data is not straightforward. It is possible that as more virtual IMU data is used, the domain shift issue becomes severe and the Random Forest classifier starts to overfit to the virtual IMU data. Adding virtual data has a detrimental effect on PAMAP2 (11-class). This follows our observations in Figure 3.6 and can be similarly explained by PAMAP2 (11-class) containing complex activities under the most dissimilar settings in comparison to the virtual IMU data.

Hence, we suggest finding the right balance between the amount of real and virtual IMU data for a model to learn the target activity pattern coexisting in both real and virtual IMU data, before overfitting to the virtual IMU data. We also anticipate that as the quality of virtual IMU improves in future versions of IMUTube, larger amounts of it will be able to be successfully integrated during HAR training.

3.3.3 Transfer Learning with Virtual IMU Data

In the previous two sets of experiments, we have demonstrated that human activity classifiers can feasibly learn from virtual IMU data, although limitations still exist. So far, we have assumed that labeled virtual and real IMU datasets for target activities are always available. In practice, such a scenario may not always be possible. For example, curating video datasets for translation into virtual IMU data could also be challenging, as titles or descriptions of videos can be arbitrarily ambiguous.

Here, we explore two additional cases for utilizing virtual IMU data: *i)* when the virtual IMU dataset contains a subset of target activity labels; *ii)* when labels for virtual IMU are not available at all. To do so, we leverage two transfer learning setups, *supervised* and *unsupervised*, respectively.

The analysis in this section represents our first attempts in utilizing more sophisticated modeling techniques from deep learning to extend the contributions of IMUTube. Our results are a first step towards handling realistic issues in label collection, as we do not yet incorporate any automated video labeling or search mechanisms. All experiments follow the same hold-out evaluation protocol detailed in Section 3.3.1.

3.3.3.1 Supervised Transfer Learning

With supervised transfer learning, we pre-train a model using labeled virtual IMU data and fine-tune it using labeled real IMU data. Importantly, the labels for pre-training and fine-tuning need not match.

We first explore the setup where virtual and real IMU data share the same set of activity labels – Imagine we have already curated video and virtual IMU data for some targeted activities, and also collected a small amount of real IMU data; instead of waiting until sufficient amounts of real IMU data is collected, we can first train a model on the virtual IMU data and fine-tune on the small-scale real IMU data. By studying this scenario, we can also gauge if pre-training with virtual data might provide any benefits to activity recognition performance.

Next, we consider a scenario where the virtual IMU data only contains a subset of the real IMU data activity classes. To examine this, we pre-train a model on the virtual PAMAP locomotion (8-classes) task and fine-tune it on the complex activities (11-classes) tasks.

Method

We compare the recognition performance of ConvLSTM models *i)* trained only with real data and *ii)* pre-trained on virtual and fine-tuned on real IMU data. The former is the same as the R2R case in Section 3.3.1. For *ii)*, we randomly split the virtual IMU data into train/validation/test ($80\%/10\%/10\%$) and pre-train the network using the virtual IMU training data. During fine-tuning, all model weights are updated and we report the performance on the real IMU test dataset; In the case where real and virtual IMU activity labels do not match, we replace the last layer of the pre-trained model with the target number of classes (thereby going from 8 to 11 activity classes for PAMAP2) and update all the network weights. We present results on all 4 datasets and follow the training and evaluation protocols described in Section 3.3.1.

Results

Table 3.6 shows the differences in F1-scores achieved by ConvLSTM models with (Supervised TL) and without pre-training (Supervised R2R) when evaluated on a random test subject. With pre-training, statistically significant performance gains are observed on Real-world and Opportunity, at 14% and 3% respectively. We further present the effects of using different amounts of real IMU data for fine-tuning the selected base model in Figure 3.7.

Dataset	R2R	Mix2R	TL
Realworld	73.1 ± 0.7	77.9 ± 0.7 (**+4.8**)	83.4 ± 0.6 (**+10.3**)
Opportunity	88.7 ± 0.7	88.4 ± 0.8 (**-0.3**)	91.0 ± 0.7 (**+2.3**)
PAMAP2 (8-cls.)	70.0 ± 1.6	70.2 ± 1.6 (**+0.2**)	71.4 ± 1.6 (**+1.4**)
PAMAP2 (11-cls.)	69.8 ± 1.3	71.0 ± 1.3 (**+1.2**)	70.2 ± 1.3 (**+0.4**)
PAMAP2 ($8 \rightarrow 11$ cls.)	-	-	70.7 ± 1.3 (**+0.9**)

Table 3.6: F1 scores of transfer learning with supervised pre-training, when evaluated on different HAR tasks. R2R is the baseline trained on real data from scratch. Transfer learning (TL) results show the performance of the models fine-tuned on real data.

Dataset	R2R*	TL
Realworld	79.2 ± 0.7	77.2 ± 0.7 (**-2.0**)
Opportunity	89.0 ± 0.7	84.8 ± 0.8 (**-4.2**)
PAMAP2 (8-cls.)	64.7 ± 1.7	68.1 ± 1.6 (**+3.4**)
PAMAP2 (11-cls.)	70.0 ± 1.3	70.0 ± 1.3 (**+0.0**)

Table 3.7: F1 scores of models learnt through unsupervised pre-training on virtual IMU data, followed by fine-tuning on real IMU data. Note that the R2R* setup here also involves unsupervised pre-training, but on real IMU data.

We see the most obvious difference in learning trajectories on the Realworld dataset, where only a small amount of real data is needed to fine-tune the base model such that it surpasses R2R performance.

In the last row of Table 3.6, we also show results for transfer learning in the case where virtual and real IMU data labels do not match (PAMAP2 8-class for pre-training, PAMAP2 11-class for fine-tuning). In this case, the model with pre-training achieves an F1-score of 0.71 on PAMAP2 (11-class), which is statistically comparable to the result in the R2R setting.

3.3.3.2 Unsupervised Transfer Learning

Unsupervised transfer learning considers the scenario where we extract the virtual IMU data from a large body of videos without labels. Curating a collection of unlabeled videos is easier relative to obtaining labeled videos, particularly in scenarios where the video descriptions/labels may be unreliable. Without a set of specific target activities in mind, any videos with humans can be utilized. Validating the feasibility of this approach represents a first step towards curating a large collection of virtual IMU data, consisting of very diverse movements and activities from which a model can learn generic representations.

Method

Our unsupervised transfer learning setup consists of two stages. The first stage pre-trains a convolutional autoencoder (CAE) to learn feature representations. The second stage ex-

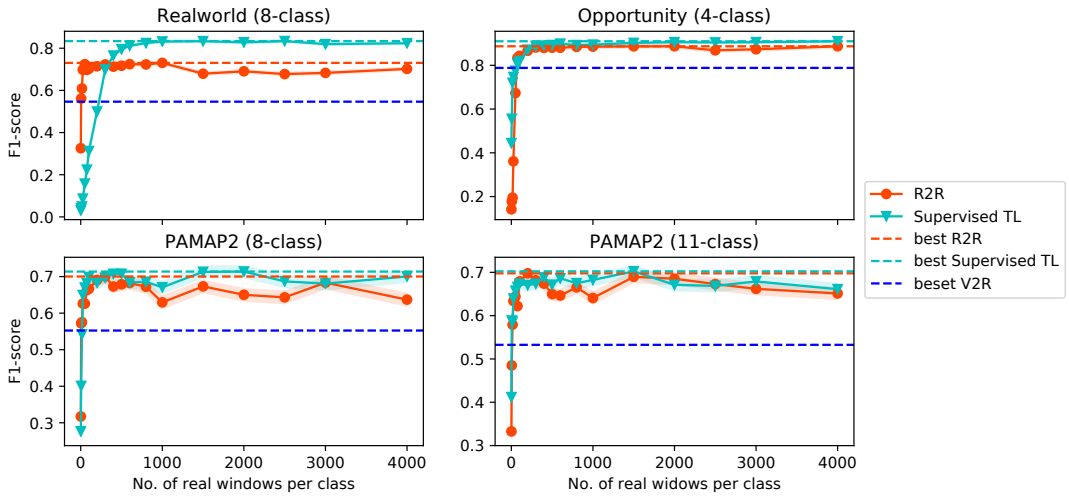


Figure 3.7: Transfer learning vs. amount of real data used for training.

tracts from real IMU data the learned representations, which are then used to train a random forest classifier. We compare the recognition performance achieved by the entire CAE-RF setup when: *i)* virtual IMU data is used for training the CAE (Unsupervised TL); and *ii)* when real IMU data is used for training the CAE (Unsupervised R2R).

We define the CAE architecture based on the model presented by Haresamudram *et al.* in [201], where the encoder contains four convolutional blocks, leading to the bottleneck layer [201]. Each block contains two 3×3 convolutional layers followed by 2×2 max-pooling. Batch normalization is applied after each layer [204]. The output from the last convolutional block is flattened before being connected to the bottleneck layer. The decoder inverts the encoder by performing convolution, interpolation, and padding to match the sizes of the corresponding encoder blocks [205]. ReLU activation [206] is used throughout, except the output, where the hyperbolic tangent function is used instead. We follow the evaluation protocol used in the ConvLSTM case.

Results

Table 3.7 shows the results for using virtual IMU for pre-training with varying performance relative to models trained on real IMU data. On Realworld, Opportunity and PAMAP2 (11-class), unsupervised TL using virtual IMU data reached up to $95\% - 100\%$ of R2R F1-scores. On PAMAP2 (8-class), we even see an increase of 5% over the R2R protocol. These results demonstrate the feasibility in utilizing virtual IMU data even in scenarios where video labels are completely absent.

3.4 Discussion

In this section, we discuss the limitations of our approach and highlight potential avenues for future work.

Our qualitative inspection of the IMUTube output and the per-class V2R results suggests that certain video characteristics may contribute to poorer recognition performance on virtual IMU data: large ego-motions, multiple moving objects and people, and occlusion.

Large ego-motions can be found in the Realworld "running" videos in which the video-taker was also running, leading to significant vertical shaking motion from the camera itself. It is possible that such vertical motions end up producing features that are very similar to those from a "jumping" motion, which may explain a higher class confusion observed between "running" and "jumping" on the Realworld and PAMAP2 (8-class) tasks. An additional factor might have come from the presence of multiple moving objects and people in the background (e.g. pedestrians) in the "running" videos. We also found that V2R models struggle more in classifying activities with similar poses which have more subtle differences in limb movements, e.g. "standing" vs. "vacuum cleaning", "sitting" vs. "ironing", as in PAMAP2 (11-class). In many "vacuum cleaning" and "ironing" videos, the subject's arm movements are occluded by objects in the scene, e.g. clothes or home furniture.

On the other hand, videos with fewer or without such motion artifacts tend to produce virtual IMU data that are well-classified under V2R. Moreover, videos featuring activities with distinctive poses and motions, e.g. cycling, are well-classified under the V2R setting. There are also many existing techniques that will allow us to further tackle motion blur ([207, 208]) and occlusion ([209, 210]). It is also possible that the future curation of video data can automatically rank videos by the presence of these undesirable features to arrive at a suitable dataset for virtual IMU data extraction.

In the current version of IMUTube, have utilized a series of off-the-shelf techniques at every step of the proposed pipeline in Section 3.2. While this supports the reproducibility of our results, it does result in limitations that impact the overall quality of virtual IMU data for HAR, as discussed above. In the following, we detail the known limitations of each step of the pipeline and present steps forward to advance this line of research:

From Vision To Pose The accurate recovery of human pose from videos is a core problem

in IMUTube. While 2D or 3D pose estimation has been widely studied by the computer vision community, these methods face limitations when videos are sourced in-the-wild, which in turn hamper the ability of IMUTube to extract realistic virtual IMU signals. Occlusions, motion blur between foreground and background objects, poor or varying lighting conditions are common sources of errors from in-the-wild videos which affect the performance of computer vision tools used in IMUTube [180], leading to distorted 3D motion outputs. *OpenPose*, the state-of-the-art 2D estimator employed in IMUTube, is susceptible to issues such as partial joint detection or the swapping of left and right when rare poses are encountered [168]. Each algorithm employed in IMUTube have different failure scenarios, and their errors will likely accumulate and hurt the quality of the final outputs. In order to counteract the challenges in processing in-the-wild videos, we expect improvements to be made in 3D motion recovery by leveraging improved pose tracking techniques that are more robust to vigorous movement, a change of scenery, the presence of multiple people, and occlusion. Specialized video stabilization strategies or camera ego-motion techniques may also address these issues [211–213].

Our current pipeline only takes RGB videos as input, though we may also explore other emerging forms of data that are collected alongside videos. For instance, depth data are becoming increasingly common and they are found to be useful as additional supervision signals in training recent computer vision models [214]; this points to future activity datasets that could potentially provide large amounts of RGB-D data, which IMUTube can use as an additional input to better constrain the estimated 3D motion and resulting output. Given its close links to problems in computer vision, we believe future extensions of IMUTube should be continually informed by the latest developments in algorithms and hardware in the vision community.

From Pose To Accelerometry Our current approach assumes an equivalence between acceleration measured by a device on the wearer's body with that measured at the nearest body joint. This view discounts any consideration of factors such as body mass, device movement, and skin friction. To better model the on-body location of IMU devices, utilizing techniques from body mesh modeling is a straightforward solution to increase realism to the pipeline. We foresee that investigating the use of body mesh might also bring up the possibilities of synthesizing credible accelerometry data from people of different body shapes from the movement of a single human skeleton pose [215–217]. In addition, while we have only considered the generation of virtual accelerometry data in this work, we can adapt most parts of the pipeline to generate the full set of IMU signals, including gyroscope and magnetometer readings.

From Accelerometry to Virtual IMU Real IMU data, which have been the basis of building HAR classifiers, are not free of noise. Sensor noise may come from factors such as drift, hysteresis, and device calibration. To carry over such characteristic sensor noise on our virtual data, domain adaptation techniques can be deployed as well as more sophisticated techniques like Generative Adversarial Networks [218, 219].

Learning from Virtual Data A domain shift exists when a machine learning model is trained from virtual data and tested on real IMU data. Again, domain adaptation strategies to the input of the machine learning model is a solution. Alternatively, it will be promising to investigate domain-invariant features learned from virtual and real data, which could potentially lead to performance gains in HAR.

3.5 Related Work

A core motivation for IMUTube is the labor-intensive nature of conventional methods for collecting activity data with IMUs, which may be overcome by translating activity data from a more readily available modality. A few previous works have also be driven by this goal to explore alternative data collection methods that do not directly involve human participants. Kang *et al.* render a 3D human model on computer graphics software and simulate human activities [131]. The sensor data is extracted from the simulated human motion, and subsequently used to train the recognition models. However, the cost of manually designing realistic, more complex human activities may remain a bottleneck. Therefore, such methods typically only explore simple gestures and locomotion activities.

Alternatively, [126, 129] extract virtual IMU data from public, large-scale motion capture (mocap) datasets [127, 128, 220], which contain a variety of motions and poses for human activity recognition. Although these datasets cover hundreds of subjects and thousands of poses and motions, they rarely include everyday activities. The majority of such MoCap datasets include dancing, quick locomotion transitions, and martial arts, which capture only a narrow spectrum of daily human activities.

Concurrent to this work, Rey *et al.* [221] proposed to generate virtual IMU data from online videos, and demonstrated the effectiveness of their generated virtual sensor data in the recognition of fitness activities. Their proposed method assumes videos that are limited to

those with a single person and captured from a fixed static camera viewpoint, while our work considers videos with multiple people and in-the-wild scenes that permit camera motion. From each video, their method computes the 2D pose motion, and learns a regression mapping to produce the target virtual sensor data using paired video and accelerometer recordings. Their method learns the change in motion from a 2D scene to produce the acceleration magnitudes, while our proposed work, IMUTube, is designed to produce the full tri-axial inertial measurements by performing 3D motion estimation. Moreover, IMUTube also does not require model training with a paired cross-modal dataset, which is expensive to obtain; this is because our method breaks down the extraction of virtual IMU signals from videos into individual steps that are solved separately using pre-trained models or pre-defined algorithms.

3.6 Summary

In this chapter we developed a framework for generating virtual IMU data based on automated extraction from videos as a means to collect large-scale labeled datasets to support research in inertial human activity recognition. We designed and validated our framework, IMUTube, which integrates a collection of techniques from computer vision, signal processing, and machine learning. Our initial findings show promising results in using virtually generated IMU data in lieu of, or in addition to, real IMU data in training competitive inertial HAR models. Overall, this chapter supports our thesis that visual information, in the form of visual *data*, can be manipulated to avail the current labor-intensive data collection and annotation process in inertial HAR. In the next chapter, we shift our focus away from visual data and instead turn to the *modelling* aspects of the visual domain, and propose ways to exploit its learned models to benefit the learning of inertial HAR models.

Chapter 4

Transferring Knowledge from Visual to Inertial Activity Recognition

4.1 Introduction

The need to expose inertial HAR models with large amounts of activity data motivated the introduction of a video-to-IMU data translation pipeline in the previous chapter. The pipeline, IMUTube, allows us to scale up training datasets for inertial HAR models by generating virtual IMU data that depict realistic and diverse motions. With the approach, existing large-scale labeled datasets curated for visual HAR tasks can be readily converted into their IMU counterparts for inertial HAR.

An orthogonal approach to using video HAR datasets is to exploit their learned *models* – high-performing neural networks trained to classify human activities given video data as input. We ask, is there something generic in the classification of human activities that can be learned on videos and then transferred to enrich IMU representations for inertial HAR? In this chapter, we turn to the problem of cross-modal knowledge transfer, with the goal of transferring knowledge encoded in state-of-the-art HAR models trained from visual data to enrich inertial HAR models. We intuit that third-person visual data provides additional knowledge about global body movement that will complement the targeted motion patterns provided by IMU data, and as a result learn better embeddings for inertial HAR through visual-to-inertial knowledge transfer.

While we draw inspiration from cross-modal knowledge distillation studies conducted in areas outside of inertial HAR (background summarized in Section 2.2.2), we identify three

main challenges which preclude a straightforward application of current techniques and call for non-trivial adaptations to be made. First, knowledge transfer is typically conducted between source and target modalities within the visual domain, such as RGB, depth, optical flow, thermal data. An arguably more challenging situation of knowledge transfer involving visual and IMU data, where a large information gap exists, is less explored. Second, the majority of previous works assume paired multi-modal data. This restriction has limited utility in the context of visual and IMU data, which are rarely available together. Our third and final challenge involves a subtler observation – we face heterogeneous label spaces in our source and target modality – the class labels in existing video and IMU HAR datasets differ greatly and it is difficult to objectively construct a one-to-one mapping between them. How to leverage unpaired multi-modal data with heterogeneous label spaces remains unclear.

To tackle the above limitations, we introduce CrossHAR– a novel framework dedicated to transferring knowledge from visual to inertial domain for inertial HAR. CrossHAR builds upon a generalized distillation framework which views visual data and features as side information at training time, and learn an additional mapping which hallucinates these visual features from IMU data. With CrossHAR, we demonstrate that further increase in the performance of inertial HAR systems *without collecting extra IMU data or paired video data* can be achieved by cross-modal knowledge transfer with the following innovations: 1. focusing on only motion-related information from visual data, 2. training labeled IMU data with unlabeled video data, 3. using a cycle-consistent associative loss function. As a result, CrossHAR is able to optimally leverage the knowledge from learned video-trained HAR models to train a stronger inertial HAR model. A key consequence of CrossHAR is that it enables the learning of enriched, visually-grounded inertial representations, leading to performance gains in test-time inertial HAR performance.

4.2 Knowledge Transfer from Videos to Inertial HAR

4.2.1 Problem Formulation

Our goal is to learn an enriched inertial HAR model by tackling a knowledge transfer problem from the *video* domain to the *IMU* domain, defined as follows:

- **Video** We have a large dataset $\mathcal{D}^V \coloneqq \{x_i^V\}_{i=1}^{N_V}$ of N_V video samples depicting human activities, with dimensions $\mathbf{x}^{\text{Video}} \subseteq \mathbb{R}^{F_V}$. We also have available a state-of-the-art activity classification network, $\mathcal{M}^{\text{Video}}$, which was pre-trained using a large annotated video dataset for C_V-class HAR task.

- **IMU** We have a dataset $\mathcal{D}^M = \{(x_i, y_i)\}_{i=1}^{N}$ of N annotated samples, where each sample is a low-dimensional inertial data sequence $\mathbf{x} \in \mathbb{R}^F$. The samples belong to one of C activity classes. We also have a randomly-initialized generic base architecture $\mathcal{M}$, which is appropriate for the C-class classification task specified by $\mathcal{D}^M$.

The discrepancy of resources in the two domains can be seen in terms of dataset size ($N_V >> N$), data dimensionality ($F_V >> F$) and label space ($C_V >> C$), making *video* a suitable source domain from which to transfer knowledge. Easy access to all of the above resources is assumed during model training, but not at test time. As such, the knowledge transfer task can also be viewed as a problem of learning with side information, which in this case refers to the video data and their learned representations. At test time, the model only sees inertial data, from which it will extract enriched representations for activity classification.

4.2.2 Architectural Overview

We consider the case of producing a test-time inertial HAR classifier, enriched through learning from both video and IMU data at training time. To accomplish this, we propose CrossHAR.

Our formulation of CrossHAR is given in Figure 4.1 and it works as follows: At training time, unpaired video and IMU data are available. The video data are processed through a pre-trained visual feature encoder to output rich visual features. We cause the visual modality to share information with the inertial modality by instantiating a dual-branch network that takes IMU data as input: an inertial branch is learned using the standard supervision classification loss, and a cross-modal branch is additionally guided by a distillation loss to mirror the visual features. At test time, only the dual-branch network is kept for making predictions on IMU data.

Our motivation for a dual-branch network is to supplement the conventional inertial encoder with cross-modal features that are grounded in the visual embedding space. Its structure is inspired by modality hallucination networks [137], which also adopt a dual-branch approach but rely on paired multi-modal data for training.

To limit the information gap between features extracted from videos and IMU data, we restrict our consideration of videos to their motion-related content only. We are motivated to do so because IMU data alone is insufficient to account for appearance-related video content such as lighting, background, and texture. This can be easily accomplished by choosing an appropriate visual encoder that encodes motion-based features, instead of appearance-based ones.

4.2.3 Training CrossHAR

We now provide a detailed description of a forward pass through CrossHAR and how it is optimized during the training phase.

At training time, CrossHAR consists of: 1. the visual encoder $f(\cdot)$, which denotes the feature encoder of the pre-trained video encoder $\mathcal{M}^{\text{Video}}$, 2. the Inertial branch $\mathcal{M}^{\text{Inertial}}$, and 3. the Cross-modal branch $\mathcal{M}^{\text{Cross-modal}}$. $\mathcal{M}^{\text{Inertial}}$ and $\mathcal{M}^{\text{Cross-modal}}$ are randomly initialized with the same architecture $\mathcal{M}$, and their feature encoders are denoted respectively as $h(\cdot)$ and $g(\cdot)$. Feature encoders refer to all layers up to the penultimate layer of each network, and for simplicity we define the branches such that the output dimensions of $f(\mathbf{x}^{\text{Video}})$, $g(\mathbf{x})$ and $h(\mathbf{x})$ are equal. The weights of g and h are updated at training time while those of f are kept fixed.

We adopt an unpaired training scheme as follows. Each IMU sample $\mathbf{x}$ and its label y are randomly paired with an unlabelled, video sample $\mathbf{x}^{\text{Video}}$ to form training triplets of the form $(\mathbf{x}^{\text{Video}}, \mathbf{x}, y)$. Through the visual encoder f, the video sample is mapped to a feature vector $f(\mathbf{x}^{\text{Video}})$. The IMU sample $\mathbf{x}$ is passed through M^{Inertial} and $M^{\text{Cross-modal}}$ to produce separate predictions, which are averaged to produce the following classification loss:

$$\mathcal{L}_{\text{classification}} = \mathcal{L}_{\text{cls}}(y, \hat{y}^{\text{Inertial}}) + \mathcal{L}_{\text{cls}}(y, \hat{y}^{\text{Cross-modal}}) + \mathcal{L}_{\text{cls}}(y, \hat{y}^{\text{Mean}}) \tag{4.1}$$

where $\hat{y}^{\text{Inertial}}$ and $\hat{y}^{\text{Cross-modal}}$ are the predictions from each branch and $\hat{y}^{\text{Mean}}$ their average.

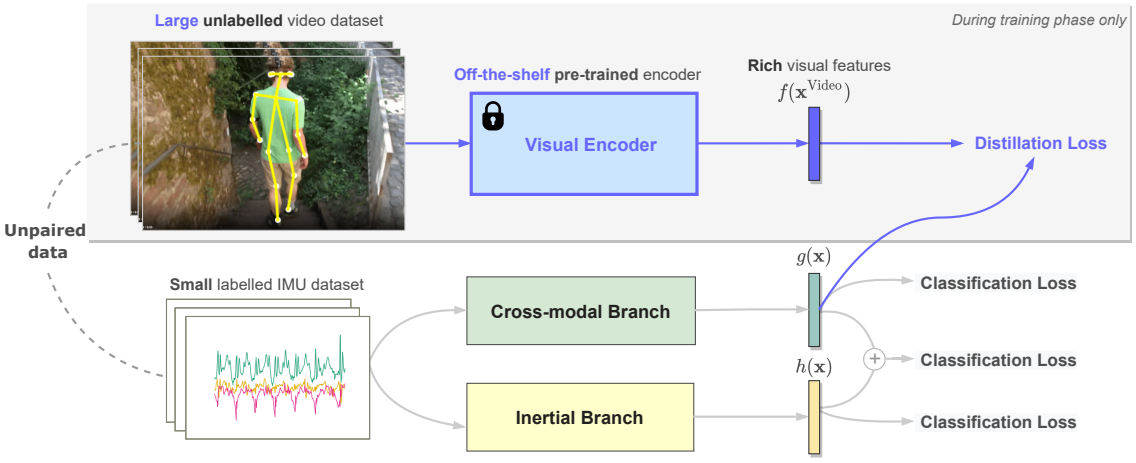


Figure 4.1: Overview of CrossHAR. An off-the-shelf, pre-trained visual encoder $f(\cdot)$ is present at training time only. The Inertial branch $h(\cdot)$ is trained by predicting correct activity class labels for an input IMU sample, while the Cross-modal branch $g(\cdot)$ is additionally guided by a distillation loss to mirror the outputs of the visual encoder. Together, the two inertial encoders learns complementary embeddings and are kept at inference time to provide rich representations of IMU data for activity recognition.

To encourage video-to-IMU information sharing, the Cross-modal branch is additionally guided by a distillation loss, $\mathcal{L}_{\text{distillation}}$, in order to minimize the latent feature discrepancy between $f(x^{\text{Video}})$ and $g(x)$. We utilize Kullback-Leibler (KL) divergence, a popular measure of the similarity between two distributions to compute the distillation loss. Later in this section, we describe a cycle-consistent variant of this loss function which accounts for the unpaired nature of video and IMU samples.

Combining the distillation and classification loss with a weighting factor α, the total loss function is given as:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{distillation}} + \mathcal{L}_{\text{classification}} \tag{4.2}$$

Training with Association Cycle Consistencies

Using forward-backward consistency, or cycle consistency, has been a standard trick for working with unpaired domains so that structure may be instilled in the unconstrained problem [154]. Following this line of thinking, we extend our distillation loss function to incorporate cycle consistency, enforced through encouraging consistent "association cycles" between video and IMU features.

Association cycles [222] provide a useful concept for relating data from a labeled domain A and an unlabeled domain B. Essentially, the inter-sample similarity is interpreted as the transition likelihood of a random walker, and association cycle consistency enforces that round-trip transitions ($A \rightarrow B \rightarrow A$) should begin and end at a sample with the same class label. This has the effect of learning embeddings that have a higher similarity if they share the same class label, thereby resulting in better model discriminative performance.

We follow this concept to define a cycle-consistent distillation loss function as follows: A refers to the labeled IMU samples and B the unlabeled video samples. Given a mini-batch of randomly paired multi-modal data, we compute similarity matrix between the dot product of their features as $M = f(x^{\text{Video}}) \cdot g(x)$. Then we calculate the IMU-to-Video and Video-to-IMU transition probability matrices by taking the softmax of M and $M^{\top}$ over columns respectively to produce $P^{\text{IMU}\rightarrow\text{Video}}$ and $P^{\text{Video}\rightarrow\text{IMU}}$; The matrix multiplication of the two gives the round-trip transition probabilities as $R = P^{\text{IMU}\rightarrow\text{Video}\rightarrow\text{IMU}}$. Finally, we calculate the distillation loss by taking the cross entropy between R and the uniform target distribution of correct round-trips T:

$$T_{ij} = \begin{cases} \frac{1}{\text{Occurrences of } y_i} & \text{if } y_i = y_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

$$L_{\text{distillation}} = H(T, P^{\text{IMU}\rightarrow\text{Video}\rightarrow\text{IMU}}) \tag{4.4}$$

We intuit that, the cycle consistency constraint provides an regularization effect on model training by encouraging embeddings which respect class distinctions among IMU samples. This is in contrast to a simple divergence measurement which does not prevent the learned features from being statistically similar but not class-discriminative. In our later empirical evaluation (Section 4.4.2), visual inspections of the transition matrices $P^{\text{IMU}\rightarrow\text{Video}}$ and $P^{\text{Video}\rightarrow\text{IMU}}$ suggest that an added benefit of allowing visually-grounded feature representations to be extracted from IMU data.

4.2.4 Choice of base networks

We now describe the specific network architectures which we adopt throughout this work for encoding video and IMU data in CrossHAR.

Video Network

In Section 4.2.2, we argue that a favorable visual encoder in CrossHAR should confine its feature space to only motion-related information so as to limit the information gap between visual and inertial features. To this end, we implement an approach that effectively breaks down the visual encoder network into two parts: 2D pose extraction followed by pose-based activity recognition.

Our visual encoder thus works as follows: Given an input video, we first use the OpenPose toolbox [109] to extract 2D joint locations for 18 body points on each video frame, keeping up to 2 people per frame and discarding noisy frames with ≥ 3 uncertain joint positions. The extracted pose arrays are subsequently passed to Spatial Temporal Graph Convolutional Networks (ST-GCN) [223], instantiated with pre-trained weights on the Kinetics-400 task. This completes the visual encoding part of CrossHAR, resulting in visual feature arrays with $h_{\text{Visual}} = 256$ dimensions. Both models (OpenPose and ST-GCN) are easily accessible models that achieve state-of-the-art results in carrying out the respective task.

Inertial Network

We adopt a simple convolutional network as the base inertial HAR network architecture (used for each of the Inertial and Cross-modal feature branch). Its architecture is similar to the Transformation Prediction Network (TPN) employed in [224, 225], which has been shown to be effective for learning temporal structures in inertial sensor data. Our inertial network consists of three 1D convolutional layer of 32, 64 and h feature maps with kernel sizes of 24, 16 and 8 respectively, each has stride 1 and dropout is applied between all layers. We set $h = h_{\text{Visual}} = 256$ for ease of feature matching between $f(\cdot)$ and $g(\cdot)$. The final layer is a single fully-connected layer with C output units. We use the standalone TPN as our "IMU-Only Baseline" throughout our empirical evaluation, it is trained using only IMU data and compared against CrossHAR (trained using visual and IMU data).

4.3 Experimental Setup

In this section we describe the data and our model implementations for subsequent empirical evaluations.

4.3.1 Datasets and Tasks

In Table 4.1, we document the statistics for all dataset considered in this study and describe them below. We define m as the video-to-imu data ratio, which is the number of training video samples divided by the number of training and validation IMU samples.

Video Data Many large-scale video datasets of human activities already exist in the public domain [24, 72, 73, 75, 77, 226]. Amongst these we downloaded the Kinetics dataset [24] and a random subset of STAIR classes [77] to guide the assembly of the video dataset used in this work. In addition, we also use videos provided in RealWorld (which also contains concurrent IMU data). We adopt similar video pre-processing steps adopted in [223], which included lowering the video resolutions and assuming a frame rate of $30Hz$. This results in video samples of dimensions $(T_v, h, w, 3)$, where T_v is the number of time steps and set as 300 on Kinetics and 150 on STAIR and RealWorld, and $h = 340$ and $w = 256$ are the re-scaled dimensions of the input video. We use all obtained samples in the training phase of CrossHAR. In practice, since our approach only involves a forward pass of video clips through the video encoder with fixed weights, the video features are pre-computed and only called during model training to increase efficiency.

Realworld Dataset Our primary evaluations are performed on the Realworld dataset (introduced in Section 2.1.1.2), which provides both IMU and video data. We consider a HAR task in Realworld for classifying 50Hz acceleromter data collected from the forearm position into one of 8 activity classes correctly, which consist of postures (e.g. "sitting") and locomotions (e.g. "climbing up").

The utility of CrossHAR lies in its use of easily obtained video data. Following this line of thinking, we should only use video data that are *not* concurrently recorded with the IMU data for model training. In particular this means that CrossHAR models evaluated on Realworld will not be trained using the video data from the same dataset but will only use video data from Kinetics and STAIRS. We evaluate the impact of this decision in 4.4.3.

Our focus on Realworld is motivated by its provision of concurrently-recorded data from both modalities, and also because a high video-to-IMU ratio is available give our available video data ($m = 25.5$). These factors together give us ample room to study the impact of varying data conditions in our empirical evaluation (Section 4.4.3).

Other IMU Datasets We provide additional results on two other HAR benchmark datasets to validate our approach, namely HHAR and WISDM (also introduced in Section 2.1.1.2).

	Dataset	Users	Classes Used	Video Samples *Train*	IMU Samples Used *Train*	*Valid.*	*Test*	IMU Position	$m = \frac{\#\text{video}}{\#\text{IMU}}$
Video	Kinetics	-	400	260,232	-	-	-	-	-
	STAIRS	-	27	2,851	-	-	-	-	-
IMU + Video	RealWorld	15	8	6,086	10,310	2,578	3,518	forearm	25.5
IMU	WISDM	51	16	-	20,314	3,302	3,324	wrist	13.3
	HHAR	9	6	-	35,539	8,885	11,959	waist	7.6

Table 4.1: Statistics of IMU and Video data used in this work. The last column displays m, the video-to-IMU data ratios when employing CrossHAR on RealWorld, WISDM and HHAR.

For HHAR [42], inertial data are also collected from activities spanning postures and locomotions. The task considered in HHAR is a 6-class activity classification using accelerometer data collected from different smartphone models worn around the waist (recorded in frequencies ranging from 50Hz to 200Hz). The WISDM dataset [51] covers a wider variety of activities, including postures, locomotions, eating (e.g. "eating pasta") and household activities (e.g. "brushing teeth"). We consider a 16-class activity classification using 20Hz accelerometer data collected from user's wrist.

Train-Test Splits We split each dataset by users so that all reported results are evaluated on users unseen during model development. Data from approximately 20% of users in each dataset are reserved as *Test Sets*. The remaining users are then split randomly by sample in $4:1$ ratios into *Train* and *Validation Sets*.

Data Pre-processing We take reference from [225] to apply minimal pre-processing steps on the raw IMU data, which begins with z-normalization of the data using the mean and standard deviation of the training set. We then apply sliding windows with 50% overlap to obtain the final samples. Except WISDM, the resulting samples have dimensions of $(400, 3)$, where 400 is the number of time steps (ranging from 2 to 8 seconds) and 3 are the tri-axial acceleration values. Due to the low sampling rate of WISDM ($20Hz$) we halve the window length such that each sample has dimensions of $(200, 3)$.

4.3.2 Model Implementations

All machine learning models are implemented in PyTorch [227] and optimised using Adam [186]. All models are trained for a maximum of 30 epochs. To obtain strong IMU-Only baselines, we performed 200 random hyperparameter trials while varying the learning rate

from 10^{-6} to 10^{-3}, L2 regularization from 10^{-8} to 10^{-4} and dropout from 10^{-3} to 0.5. These hyperparameters are reused for CrossHAR models, for which we additionally perform a grid search of α from 0.05 to 10. We report macro F1-Score as it is the most common metric in inertial activity recognition. We average 3 random seed initializations in reporting the F1 scores alongside their standard deviations.

4.4 Experiments

In this section, we present an in-depth empirical evaluation of CrossHAR on Realworld. We perform experiments to understand sources of performance gains as well as the impact of data availability on HAR performance. Finally, in Section 4.4.4 we validate the general performance of CrossHAR by providing its HAR results across two other benchmark datasets.

4.4.1 Activity Recognition Performance on Realworld

In Table 4.2, we show the F1 scores of our framework on Realworld, with two different loss functions compared to models trained only with IMU data. For each loss function choice we first compare against the IMU-only baseline (top row). A first observation is that training inertial HAR models using both IMU and video data under the CrossHAR setup provides tangible performance boosts. The increase in F1 scores brought by CrossHAR (Asso.) models over baseline amounts to 17% in relative terms. The success of this setup can be attributed to the fact the number of total training instances increased by almost an order of magnitude across three datasets, even though the added instances came from unpaired, unlabeled video data.

We also observe that training CrossHAR using either loss functions (with and without enforcing cycle consistencies) is able to provide improvements over baselines. The success seen in training CrossHAR with the KL function alone suggests that the IMU-extracted Cross-modal features provided generic representations which are already complementary to the IMU features, regardless of its activity class.

	Training Data	Distillation Loss	RealWorld
Baseline	IMU	-	64.0 ± 1.2
Ensemble Baseline		-	66.8 ± 0.0
CrossHAR	IMU + Video	KL	73.6 ± 3.7 (**+9.6**)
		Asso.	74.7 ± 2.5 (**+10.7**)

Table 4.2: F1-Scores for CrossHAR and baseline models evaluated on three HAR benchmarks. The absolute differences between each CrossHAR model and IMU-Only baseline (top row) are provided in brackets.

Since our test-time model consists of two branches, we additionally compare our approach to an ensemble baseline which also consists of two branches. We find that our CrossHAR-trained models still offer the highest performance overall. This confirms that our model offers more benefits than a simple ensemble.

Confusion Results

We inspect the class-specific confusions made by the baseline model compared to the CrossHAR (Asso.) model in Figure 4.2. We observe that our approach was able to reduce a number of great class confusions which exist in the IMU-Only setup. This is especially noticeable in "walking" vs. "climbing up", which the IMU-Only baseline model confuses the most. We believe that CrossHAR encourages the encoders to learn domain-invariant features which are less tuned to the specifics of the seen users. We also note the large confusion between "jumping" and "standing" on the IMU-Only baseline, which may be attributed to data imbalance ("jumping" is the most imbalanced class accounting for one-fifth of the other classes on RealWorld); The reduction in its confusion on the CrossHAR model may be related to the substantially larger training data size that our approach was able to incorporate through use of video data, suggesting its benefits even in solving common challenges like data imbalance.

4.4.2 Are enriched feature representations being learnt?

We have thus far seen that test-time inertial HAR performance improves by using CrossHAR, which suggests IMU feature representations are enriched through cross-modal knowledge transfer. We now seek to investigate how this has occurred through the lens of cross-modal transition probabilities introduced in Section 4.2.3. Specifically, we analyze the features

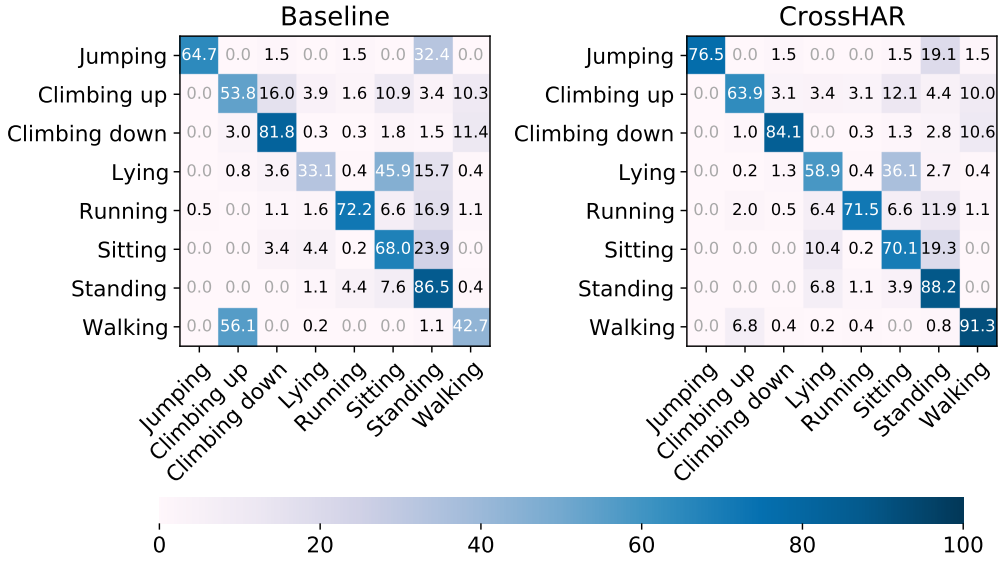


Figure 4.2: Normalized confusion matrices on Realworld when training with baseline and CrossHAR models. We show that training with videos results in large reductions in class confusions, with double-digits improvements in some cases.

extracted by the inertial feature encoder $g(\cdot)$ with and without knowledge transfer, and compute their transition matrices with respect to the video features. Since the transition probabilities are an indication of the similarity between video and IMU features, monitoring their differences will reveal how IMU feature representations have changed with respect to that of videos (which are fixed).

Visualizations of transition matrices on two random batches are displayed in Figure 4.3. The transition matrices are computed between visual features, extracted through the visual encoder with fixed weights, and IMU features extracted by the encoding function $g(\cdot)$ initialized with weights from: the IMU-Only baseline model (a and c), and the CrossHAR (Asso.) model (b and d). Darker lines denote a higher similarity (softmaxed dot product).

Looking at the baseline transitions (a and c), we see that encoder trained only with IMU data is already able to make some naive associations based on the produced embeddings. However, we observe a hubness issue where, many IMU samples, despite being of different classes, collapse into the same video samples – those most similar to themselves. This indicates that, with respect to the video embeddings, the IMU features lie close to each other. In contrast, IMU features from CrossHAR form transitions with all video samples, but the most dominant transitions (i.e. the darkest lines) respect class distinctions. In general, only IMU samples of the same class are observed to be associated with any video sample, which

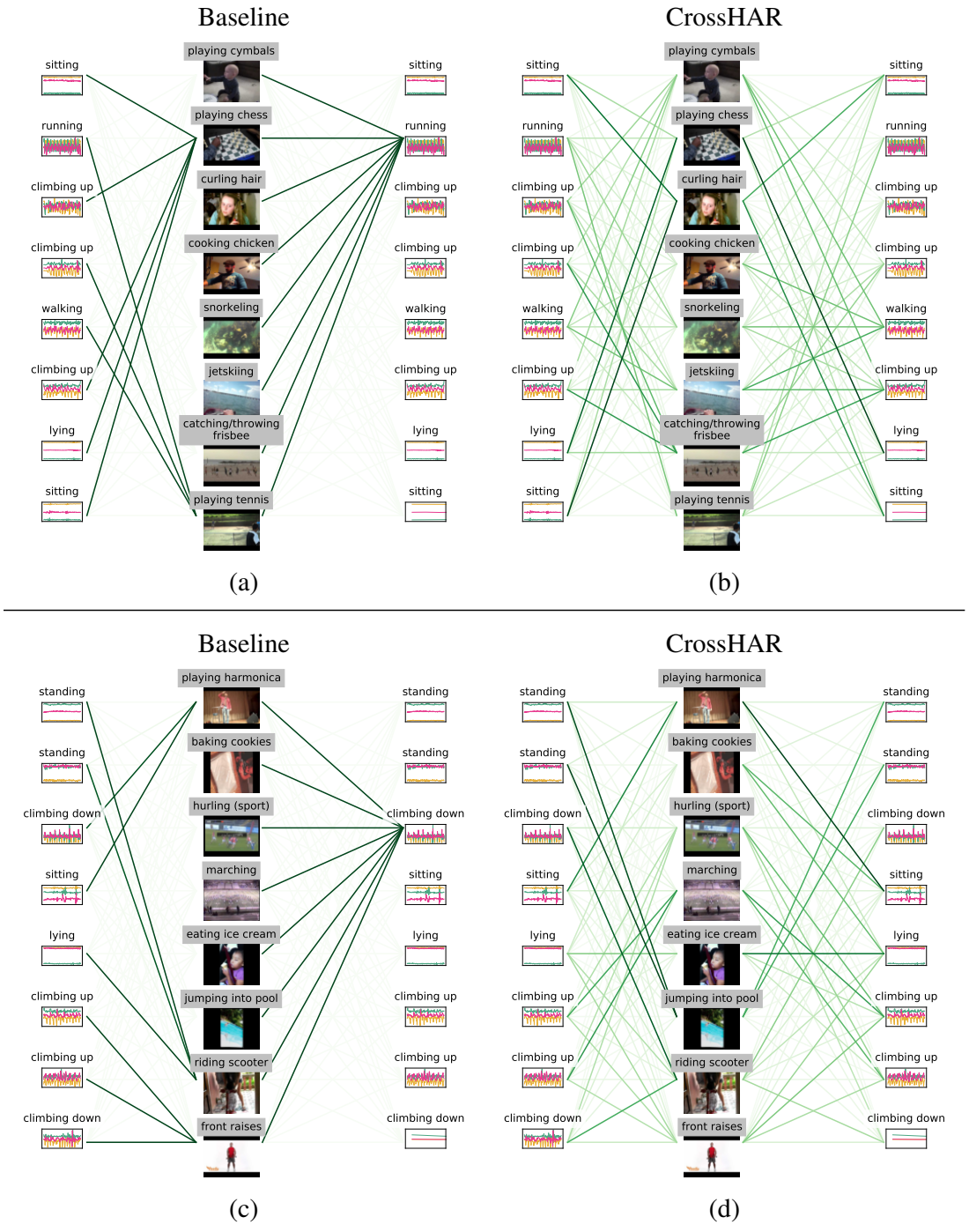


Figure 4.3: Visualization of cross-modal transitions, darker lines denote higher similarity (and thus associations): (a) visualizes cross-modal transition probabilities between IMU and video samples in a random batch, with IMU features computed by the IMU-only baseline without any knowledge transfer; Its left half depicts $P^{\text{IMU}\rightarrow\text{Video}}$ and the right $P^{\text{Video}\rightarrow\text{IMU}}$. (b) shows these transition probabilities after knowledge transfer on CrossHAR, which are now grounded within the visual embedding space – IMU samples with different class labels no longer congregate into the same video sample but instead are associated with distinct video samples, suggesting that the inertial features have become more class-discriminative after knowledge transfer. (c) and (d) illustrate a similar case by showing the baseline and CrossHAR transitions on another random batch.

means they are forming consistent association cycles. For example, “sitting”, “climbing up” and “lying” IMU samples all transition to the “playing chess” video sample in (a), but their dominant associations are formed with distinct video samples in (b). On the whole, these changes reflect that, after knowledge transfer, IMU features are much more integrated within the video embedding space, and video features are akin to anchors which guide IMU features to become more class-discriminative.

Interestingly, we find that many of the strong cross-modal transitions made by CrossHAR also seem to make intuitive sense as we compare their activity labels, e.g. “sitting” and “playing chess”, “climbing up” and “marching”. This suggests that the learnt associations may carry useful information about the natural groupings or hierarchies in activities, as some of them seem to characterize the activities’ energy expenditure or denote their compositions.

4.4.3 Impact of Training Data Availability

The effectiveness of CrossHAR is predicated on the availability of visual and IMU data at training time. We now study how realistic changes in this data availability may affect the test-time performance of CrossHAR. Throughout this subsection, we present an evaluation on Realworld comparing the performance of CrossHAR trained using the associative loss function against IMU-only baselines.

How does performance vary as video and IMU data pairing is changed?

The first question we investigate is if the quality of knowledge transfer changes due to the use of concurrently-recorded video data for paired training. To understand this we compare the following three conditions:

1. **Paired training using concurrently-recorded IMU and video data** The training dataset consists of IMU and video training samples from the Realworld dataset. Since a number of video samples are discarded after pre-processing, the video-to-IMU data ratio works out to be 0.6. In contrast to unpaired training (used across all other evaluations in this chapter), video and IMU samples are paired based on their class labels during model training.

2. **Unpaired training using IMU data and online videos** The training dataset consists of IMU training samples from Realworld and video samples downloaded from Kinetics and STAIRS (referred to as "online videos"). This represents the setup aligning the most with our vision for CrossHAR, and is the setup used in all evaluations on Realworld throughout this chapter.

3. **Unpaired training using IMU data, concurrent videos, and online videos** The training dataset consists of IMU and video training samples from Realworld, in addition to the downloaded online videos. This represents the occasion where practitioners, besides online videos, also have access to video data recorded during IMU data collection.

Naturally, these three conditions result in different training dataset sizes. We thus compare them from two angles, one where we hold the data size constant (by taking a random subset of the larger datasets) and another where we do not. We report the results in Table 4.3.

Holding the amount of video data constant (at $m = 0.6$), paired training with concurrent videos indeed performs the best, which is consistent with our expectations since this most resembles typical cross-modal transfer setups leveraging paired data. We highlight that this is also the most costly and unlikely setup due to its need for co-occurring IMU and video data, and indeed this setup naturally gives us the smallest training dataset across the three settings.

We next compare the three data settings, this time without artificially reducing the dataset sizes of the latter two. We see that CrossHAR effectively offsets the advantage brought by paired training after using more (unpaired) video data, which can be cheaply added by downloading from online video repositories. We also observe a modest performance increase when using both concurrent and online videos in the unpaired setting, which means CrossHAR can adequately exploit both types of video.

Does using more video data lead to bigger performance gains?

The next question we investigate is if by increasing the video-to-imu ratio, m, we may see bigger performance gains by CrossHAR. We stick to the IMU + Online Videos setup, and train CrossHAR models using datasets of different sizes by randomly sampling a subset of video data to reach m values of $\{0.6, 5, 10, 15, 20, 25.5\}$.

	#video / #IMU	Source of Training Data			F1 Score
		IMU	Concurrent Videos	Online Videos	
Baseline	-	✓			64.0 ± 1.2
Paired	0.6	✓	✓		69.5 ± 2.0 (**+5.5**)
Unpaired	0.6	✓		✓	66.4 ± 2.6 (**+2.4**)
Unpaired	0.6	✓	✓	✓	66.8 ± 1.4 (**+2.8**)
Unpaired	25.5	✓		✓	74.7 ± 2.5 (**+10.7**)
Unpaired	26.1	✓	✓	✓	75.7 ± 1.0 (**+11.7**)

Table 4.3: We compare F1 results of models trained with video data from different origins, applied on RealWorld, a dataset that provides both IMU and concurrently recorded video data. Holding the amount of video data constant, paired training with concurrent videos indeed performs the best, though this is also the most costly and unlikely setup due to its need for co-occurring IMU and video data. We show CrossHAR effectively offsets such advantage brought by paired training after using more video data, which can be cheaply added by downloading from online video repositories. We highlight in grey the setting used across all our analyses on Realworld.

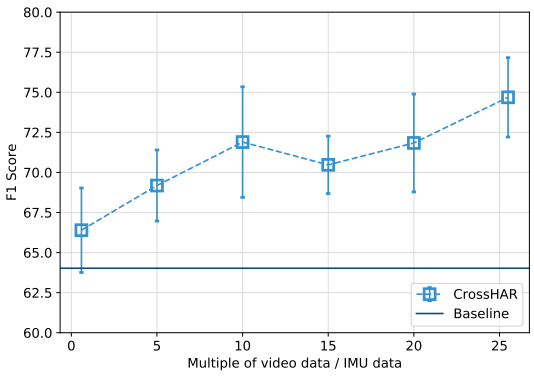


Figure 4.4: F1 results on Realworld as the amount of video training data increases, expressed as a multiple of the amount of IMU training data. We observe that training with even $0.6\times$ video data brings decent performance gains. We show that CrossHAR can exploit additional video data effectively as the level of performance gains generally increases with the amount of video data.

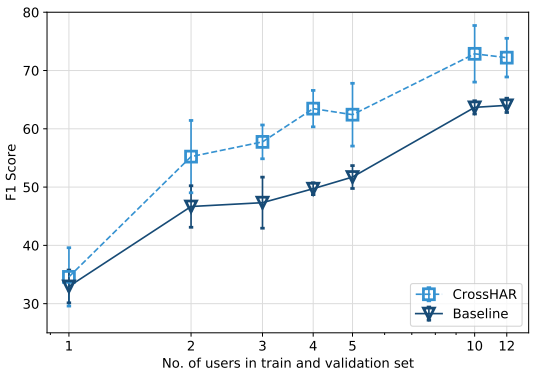


Figure 4.5: F1 results on Realworld as the amount of IMU training data increases, as varied by the number of training users. Video-to-IMU data ratio is kept at 10. We observe that IMU data from more than 2 training users is necessary for the effective training of CrossHAR, after which our method consistently outperforms baselines by a large margin.

We report the obtained F1 scores on Realworld in Figure 4.4. We see a general upward trend when transferring increasing amounts of video data. This validates our vision for using video data, which can be easily downloaded to increase m, as the source modality for knowledge transfer in CrossHAR. We see a slight drop in performance going from $m = 10$

to $m = 15$, which we believe to be an artifact of the subset of video data used for knowledge transfer. With a larger and more diverse dataset, we see that performance gains eventually pick up and increase again.

How much IMU data is needed for knowledge transfer to be beneficial?

Having investigated the optimal settings around video data, we now turn to the problem of IMU data availability. We posit that the knowledge transfer would only be effective if a rich enough IMU dataset is also available for training in order to make sense of the additional diversity provided by the video data. Here we investigate if a minimum level of IMU data needs to be available for CrossHAR performance gains to be seen, and if so, what is this minimum. To test this, we use the number of training users, i.e. users from which IMU samples are collected, as a measure of IMU training dataset diversity, and measure HAR performance of CrossHAR and the IMU-only baseline when this number is varied. We keep the video-to-IMU ratio constant at $m = 10$ for all CrossHAR models in this experiment.

We report the results in Figure 4.5. We see that IMU data from more than 2 training users is needed for the effective training of CrossHAR, after which our method consistently outperforms baselines by a large margin. In particular, F1 scores with our approach using only 5 training users are already on par with IMU-only baselines trained with data from 10 or more training users.

4.4.4 Evaluation on HHAR and WISDM

Having conducted an extensive investigation on Realworld, we now ask if knowledge transfer may also help on other HHAR benchmarks. In this subsection, we study the application of CrossHAR on the HHAR and WISDM datasets. We compare the CrossHAR models trained using the associative loss function against the IMU-only baseline in Table 4.4. We find that CrossHAR outperforms baseline by 9% on HHAR and 5% on WISDM. We believe the smaller performance gains are attributed to the drop in m on these datasets, which are approximately 8 and 13 on HHAR and WISDM respectively, which are almost half of that from the CrossHAR setup on Realworld ($m = 25.5$). Another reason for the modest gains on WISDM may also be due to the increased complexity of this dataset, which may require an even larger amount of video data for knowledge transfer to be effective. We also conducted some initial experiments where we vary video and IMU data sizes on both

	HHAR	WISDM
IMU-Only Baseline	79.2 ± 1.4	67.8 ± 1.5
CrossHAR (Asso.)	86.7 ± 3.8 (**+7.5**)	71.0 ± 1.7 (**+3.2**)

Table 4.4: F1-Scores for CrossHAR and baseline models evaluated on WISDM and HHAR. The absolute differences between each CrossHAR model and IMU-Only baseline (top row) are provided in brackets.

HHAR and WISDM and observe similar trends as on Realworld, indicating that the performance gains may be increased as we add more video data. We defer the exploration of additional video data collection on these datasets for future work. Overall our results indicate that knowledge transfer from videos provides benefits beyond the Realworld dataset and we expect that the gains from CrossHAR would only become larger if we were able to increase the amount of video data.

4.5 Discussion

In this section, we discuss the limitations of our work and point to promising extensions of our method as future work.

Matching the video and IMU datasets In the problem formulation of CrossHAR, we have adopted an unpaired training approach and excluded the use of class labels from the video samples at any point of model training. One reason for this decision is the difficulty in producing pairings of existing visual and IMU data, even if we are only pairing them based on class labels. Our inspection of vision datasets [24, 75, 77, 191, 193, 226] reveals a mismatch in class labels defined in visual and inertial HAR, especially when considering static activities. In many IMU datasets, a significant portion of data come from static activities, most commonly in form of *"standing"*, *"sitting"* and *"lying"* (on RealWorld, these classes alone represent $> 41\%$ of IMU data). From vision datasets, it is difficult to recover video clips depicting these specific static activities because class labels are typically defined by their utility, e.g. classes such as "texting" and "waiting in line" may well be a mixture "standing" and "sitting". Given the in-the-wild nature of most of the obtained video data, it is also natural that a single video clip may contain several activities, even in high-quality curated datasets [24].

Unpaired training of video and IMU samples The small gap between training with and without cycle-consistency suggests that the current association loss may not have maximized the benefits of training with the additional, unpaired, video data. We note that, other than the class labels themselves, we have not taken into account the matching of IMU and video samples along the temporal dimensions; e.g. a 5-second long video clip may have already contained more than one activity class found in IMU samples of 1 second long. Explicitly pre-processing samples so that they depict activities of the same duration, and associating features before they are temporally aggregated may be fruitful future directions. Further, it may be effective to explore adversarial training with cycle-consistency [154], though this will also increase the computational resources required for model training.

Associations between video and IMU samples A noteworthy observation we have in the empirical evaluation of CrossHAR is that, during in the knowledge transfer process, interesting associations are learned between IMU and video samples. We posit that these associations may carry useful information for other investigations beyond our current goal of improving a test-time inertial HAR network. Visualizations of the learned associations point to a visually-grounded IMU embedding space, that may be of use to model explanation or zero-shot learning tasks. However, observations of some implausible associations also suggest caution in directly utilizing these learned associations at this stage. For example, strong association cycles are formed between "standing" and "jumping into pool" in Figure 4.3(d), though we may interpret as very different activities due to their energy intensity. This may have been an artifact of the video quality, but it is possible that erroneous associations would harm the learning of the IMU features, as we also saw that the "standing" class was one of the most easily confused activities at test-time (ref. Figure 4.2). An option to improve these associations is to also provide the video with access to the video activity annotations, or even to incorporate language vectors of the text labels. Extension of our work may also examine the quality of these associations qualitatively by measuring the distances of these text labels in the language embedding space (e.g. `word2vec`). While we leave investigations in this space as future work, we believe that a systematic way to learn good associations between video and IMU samples is a promising future direction.

Video encoder weights Given our focus on knowledge transfer in the direction of video-to-IMU, we have fixed the pre-trained weights of the video feature encoder $f(\cdot)$ in CrossHAR. A potential source of performance improvement may be to allow these weights to be fine-tuned in the knowledge transfer process. An interesting investigation would also be to look at how video-based HAR performance is impacted, as beneficial knowledge transfer may occur in the opposite direction of IMU-to-video as video embeddings become grounded in

terms of their inertial properties. Apart from this, the layer at which the knowledge transfer takes place can also be investigated to locate the optimal point of transfer and understand what that signifies.

4.6 Related Work

Knowledge Transfer in Inertial HAR Our work is related to transfer learning and domain adaptation which aims to share information from one task to another. Transfer learning between deep learning architectures has been explored previously in inertial HAR in the context of adapting models to different body locations [61, 228], devices [229] and users [230–232]. The motivation of these works is to minimize the performance drop when a HAR model is evaluated in a different domain while using very limited or no annotated data from the target domain. This is different from the focus of this chapter, which is to learn richer representations (and by extension achieve performance gains) by transferring knowledge from a model trained in the source domain. Moreover, while our work deals with transferring knowledge between networks operating on very different data modalities (vision to IMU), most prior methods are focused on transferring information within the inertial modalities (e.g. a cross-modal setup investigated by Morales et al. was accelerometer $\rightarrow$ gyroscope [232]). A notable exception is Vision2Sensor [233], which proposes a visual-to-inertial knowledge transfer framework by providing vision-based labels to train an inertial HAR model; However their approach assumes that the IMU training data is collected in a camera-instrumented environment so that sensor data may be opportunistically and automatically annotated by a synchronized vision modality. In contrast, our work is predicated on an unpaired training setup and thus vastly enlarging the applicability of our approach such that it may be immediately applied to any existing IMU datasets to improve test-time HAR performance, even if it was not recorded alongside any video data.

Cross-modal Knowledge Distillation Our work is strongly related to cross-modal knowledge distillation works which deal with transferring knowledge from one modality to another. We draw inspiration from the modality hallucination architecture [137], which incorporates side information at training time, in the form of depth images and their learned representations, to improve a RGB test-time object detection model. Although effective, investigations of modality hallucination have predominantly been limited to knowledge transfer between image pairs from RGB, depth [137, 151] or thermal data [146, 147, 150],

owing to the availability of co-occurring datasets recorded by novel thermal and depth cameras [196,234]. Unlike these works, our work adopts modality hallucination using unpaired data to transfer knowledge from the visual to the inertial modality.

4.7 Summary

In this chapter, we have proposed CrossHAR, a novel approach for cross-modal knowledge distillation from videos to IMU data for inertial Human Activity Recognition. Over extensive experimental validations, we show that our method overcomes the challenging setup of unpaired video and IMU data, and successfully distills effective information from the video domain. Our results show that CrossHAR outperforms IMU-Only methods not only in terms of overall F1 scores but also in improving the worst-case performances, which signifies a more robust embedding space. Our investigation suggests that exploiting videos of human activities as well as their associated models developed in the computer vision space has great potential for improving activity recognition tasks using IMU data. Following these encouraging results, we turn to investigate novel ways of exploiting visual HAR models and their learned embeddings under a zero-shot learning setup in the next chapter.

Chapter 5

Zero-Shot Learning For Inertial Activity Recognition Using Video Embeddings

5.1 Introduction

When developing a Human Activity Recognition (HAR) system for IMU data, a typical first step is to define a set of activity classes to recognize. This determines what data to collect, how to collect them, and importantly, what activities can be recognized at test time. Once model learning is complete, the developed system generally cannot handle challenging scenarios where instances of new activities classes are encountered, in which case the system will output an uninformative *"other"* class (if included in training) or predict a wrong label belonging to the seen classes.

In reality, the scenario of encountering unseen activities is common in practical applications of HAR systems. The primary reason is that the number of human activities is large but existing datasets only cover a limited number of them [20] (most IMU datasets [40, 41, 50] contain fewer than 20 activity labels), which means users are bound to perform activities which are not currently recognizable. Although it is possible to build larger IMU datasets so that the HAR systems are trained with more diverse activities in the first place, the high costs associated with data collection and annotation remain prohibitive. Further, since the types of activities performed vary with each user depending on lifestyle, environment and occupation factors, it is hard to collect sufficient IMU instances for each specific activity. These limitations inhibit the use of HAR systems in the continuous monitoring of Activities

of Daily Living (ADLs), which is of great interest to psychology and healthcare research communities [54, 235, 236].

Zero-Shot Learning (ZSL) [156] offers a framework to solve the problem of recognizing previously unseen activities. Its development has been especially driven by applications related to images, videos, and natural language processing (NLP). The main principle behind most zero-shot learning methods is to associate seen and unseen classes through some auxiliary information, usually defined by a *semantic space*. This approach is analogous to how humans recognize unfamiliar objects – a common example given in computer vision is that we can recognize *"zebras"* without seeing one before, if we were given the information that *"zebra looks like horses with stripes"*, and that we can recognize *"horses"* and *"stripes"*. Therefore, the semantic space will need to contain rich information about both seen and unseen classes, and be related to the feature space. By using such a semantic space, ZSL methods can determine the class labels for instances of unseen classes, even when no labeled instances were available.

In the field of Activity Recognition from IMU data, there have been few prior studies on zero-shot learning [101, 237–239]. Most studies in this direction adopt a hand-engineered attribute space, one of the most widely used semantic spaces in zero-shot learning. However, while attributes can be intuitively defined for tasks like animal recognition (e.g. attributes could be color or habitat), defining attributes for activities are difficult since the way activities are performed may change considerably across individuals and time. As a result, domain expert knowledge is required to define the attribute space, which may look entirely different under different studies. Using a learned semantic space facilitates a principled approach to refining the zero-shot learning problem.

Although there have been attempts to employ learned semantic spaces built from word embeddings of class labels or descriptions, they have been found to give varying performance compared to manual attributes [238, 239]; This is unsurprising given the semantic gap between words and IMU signals (since words lack motion-specific information), as well as confusions when the activities have compound names [240].

In this chapter, we propose a novel strategy to construct an informative semantic space for inertial activity recognition using videos of human activities. With this strategy, feature representations of videos are extracted from state-of-the-art video action recognition models trained from thousands of videos and hundreds of activity classes. These rich feature representations, which we refer to as *video embeddings*, are then used to construct a se-

mantic space to relate seen and unseen classes. Not only does this video-based semantic space circumvent the need to manually define attributes, but it also manifests the transfer of knowledge from video-based HAR models to an inertial HAR problem.

5.2 Proposed Approach

In this section, we detail the design of our proposed video-based zero-shot learning approach.

5.2.1 Problem Formulation

Zero-shot learning tackles the problem of classifying classes that have no labeled instances available. Conforming to the notations introduced in Section 2.2.3, we denote the set of seen classes as $\mathcal{S} = \{c_i^s\}_{i=1}^{N_s}$ and the set of unseen classes as $\mathcal{U} = \{c_i^u\}_{i=1}^{N_u}$, where $\mathcal{S} \cap \mathcal{U} = \emptyset$. During training, we are given the training dataset $D_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{S}$, where N is the number of training instances, $\mathcal{X} \in \mathbb{R}^d$ is the feature space, and all instances are from the seen classes $\mathcal{S}$. During testing, we are given $D_{test} = \{\mathbf{x}_i\}_{i=1}^{N_{test}} \in \mathcal{X}$, and these test instances belong to the unseen classes $\mathcal{U}$.

Since the classifier cannot learn from any labeled instances belonging to the unseen classes, an auxiliary semantic space needs to be defined to solve the zero-shot learning problem. In the semantic space, every (seen and unseen) class has one corresponding vector representation called a *class prototype*. We denote the semantic space as $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{N_s+N_u}$, where $\mathbf{p}_k \in \mathbb{R}^F$ is the prototype for class c_k.

We adopt a projection-based approach for zero-shot learning, where the main idea is to obtain labeled instances for the unseen classes by projecting the instances from the feature space $\mathcal{X}$ onto the semantic space $\mathcal{P}$.

For an instance $\mathbf{x}_i$, the projection function $\theta(\cdot)$ is defined as follows:

$$\mathcal{X} \rightarrow \mathcal{P} : \mathbf{z}_i = \theta(\mathbf{x}_i) \tag{5.1}$$

After projection, classification is performed in the semantic space. Owing to the limited number of labeled instances in the semantic space, we use a nearest neighbor classifier (1NN) to output the predicted class by locating the closest prototype which belongs to $\mathcal{U}$.

We summarize the overall workflow as consisting of three stages: First, before model training can begin, we need to define a visually-based semantic space $\mathcal{P}^{\text{Video}}$, consisting of prototypes that act as mapping targets for inertial HAR feature instances. Then, at training time, we learn the projection function $\theta(\cdot)$ using D_{train} and $\mathcal{P}^{\text{Video}}$. Finally, at testing time, testing instances from D_{test} are projected onto $\mathcal{P}^{\text{Video}}$ using $\theta(\cdot)$ and prediction is made via 1NN classification. We give further details to each of these stages in the following subsections.

5.2.2 Constructing the Video Semantic Space

We describe our strategy for constructing the video semantic space. This assumes the availability of video data belonging to activities described in the seen $\mathcal{S}$ and unseen classes $\mathcal{U}$ (we describe the collection of video data in Section 5.3.2.)

We extract the video embedding prototype $\mathbf{p}_k \in \mathbb{R}^F$ of an activity as the set of features of this activity's video extracted with a pre-trained video HAR model M:

$$\mathbf{p}_k = \mathcal{M}(\mathbf{v}_k), \tag{5.2}$$

where $\mathbf{v}_k$ is the raw video data of activity class c_k. If multiple videos are collected for each activity class, then we define the prototype as the mean of individual feature vectors:

$$\mathbf{p}_k = \frac{1}{N_k^{vid}} \sum_{n=1}^{N_k^{vid}} \mathcal{M}(\mathbf{v}_k^n) \tag{5.3}$$

where N_k^{vid} is the number of available videos of activity class c_k.

In our experiments, $\mathcal{M}$ is defined as video activity recognition model I3D [241]. Specifically, we use an I3D model pre-trained on Kinetics-400 [24]. The Kinetics-400 dataset is a commonly-used benchmark for video activity recognition, containing $306,245$ labeled videos in total for 400 activity classes, and I3D is a popular architecture achieving state-of-the-art results on the dataset. The I3D architecture employs a 3D-CNN to treat the temporal dimension in videos as an extra spatial dimension. There are two streams in the I3D architecture, namely the RGB stream and the optical flow stream. The RGB stream inflates an

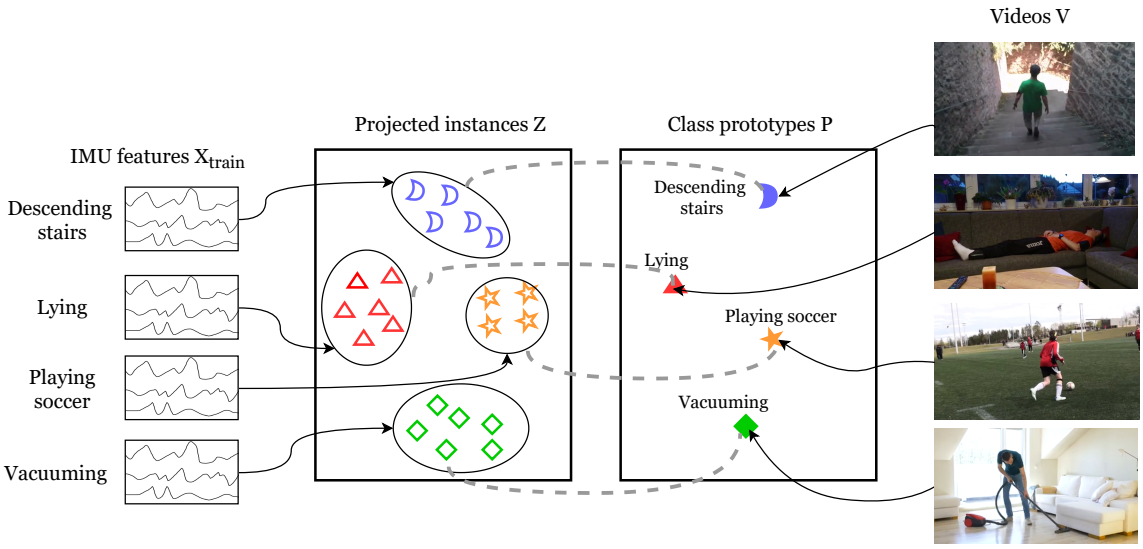


Figure 5.1: Overview of our projection-based method for zero-shot learning. The right side illustrates the construction of a video semantic space, which is done by passing video data through a pre-trained I3D model to compute class prototypes. The left side shows the projection of IMU features into the video semantic space, done using a 4-layer MLP.

Inception-v1 [242] architecture by endowing 2D filters with a third dimension and copying weights pre-trained on ImageNet [138] across this third dimension as initialization. The extra optical flow stream shares the same architecture, but in our pipeline, in the interest of efficiency, only the RGB stream is kept.

The construction of the video embedding space is represented on the right side of Figure 5.1. Videos of each activity class are passed into the RGB branch of the aforementioned I3D model, and a forward pass is conducted to extract features from the penultimate or last layer, which we refer to as *I3d-features* or *I3D-logits* respectively. Embedding vectors from *I3D-feature* has dimension $F = 1024$, and they can be seen as the output of the I3D feature extractor, encoding high-level features which are most useful for differentiating activities from video data. Vectors from *I3d-logits* are the outputs of the last layer before the final activation layer, thus giving the probability distribution over activities in the Kinetics 400 dataset, with dimension $F = 400$.

5.2.3 Projecting IMU Features into Video Semantic Spaces

In order to relate the IMU feature space and video semantic space, we project the IMU feature vectors into the extracted video embedding space. This projection operation is

illustrated on the left side of Figure 5.1. Given our focus on comparing different semantic spaces, we set up our projection function by following closely that described in [238].

The input to the pipeline is the IMU features $\mathbf{x}_i$ extracted from raw IMU data using the sliding window method. We define the projection function $\theta(\cdot)$ as a 4-layer Multilayer Perceptron (MLP) that is similar to that used in [238] and apply batch normalization [243] between all layers of the model. Each of the first 3 layers in the MLP $\theta(\cdot)$ is a typical fully connected layer with ReLU activation defined as

$$\mathbf{h}_l = relu(W_l \mathbf{h}_{l-1} + b_l), \tag{5.4}$$

where l is the layer number, W_l is the weight matrix and b_l is the bias. The MLP has two skip connections connecting from the first and second layer to the third layer, so the definition of the fourth (output) layer is

$$\mathbf{h}_4 = relu(W_4[\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] + b_4). \tag{5.5}$$

The reason behind these skip connections is that the output of previous layers may also help the classification [238]. Finally, the output of the whole MLP $h(\cdot)$ is the projected instance

$$\mathbf{z}_i = \theta(\mathbf{x}_i) = \mathbf{h}_4. \tag{5.6}$$

After projection, classification is performed with a 1NN classifier, which uses the cosine similarity as the distance metric between the projected instance $\mathbf{z}_i$ and any class prototype $\mathbf{p}_k$:

$$SIM(\mathbf{z}_i, \mathbf{p}_k) = \frac{\mathbf{z}_i \cdot \mathbf{p}_k}{\|\mathbf{p}_k\|_2}, \tag{5.7}$$

The Softmax probability of $\mathbf{x}_i$ belonging to a class c_k can thus be written as:

$$p\left(y_i = c_k \mid \mathbf{z}_i\right) = \frac{\exp\left(SIM\left(\mathbf{z}_i, \mathbf{p}_k\right)\right)}{\sum_{q \in \mathcal{T}} \exp\left(SIM\left(\mathbf{z}_i, \mathbf{p}_q\right)\right)}, \tag{5.8}$$

where $\mathcal{T}$ is $\mathcal{S}$ during training and $\mathcal{U}$ during testing.

This gives the final prediction as

$$\hat{y}_i = \underset{c_k \in \mathcal{T}}{\arg\max}\, p\left(y_i = c_k \mid \mathbf{z}_i\right), \tag{5.9}$$

5.2.4 Learning the Projection Function

To train the projection mapping, we adopt the Cross Loss function introduced in [238]:

$$L = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{p}^{y_i}\|_2 - \mu \sum_{i=1}^{N} \sum_{k \in \mathcal{S}} y_i^k \log\left(p\left(y_i = c_k \mid \mathbf{z}_i\right)\right) \tag{5.10}$$

where $\mathbf{p}^{y_i}$ is the prototype of ground truth class y_i and y_i^k is the k^{th} entry of the one-hot representation of y_i. The terms in the loss function are explained as follows:

- The first term $\sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{p}^{y_i}\|_2$ denotes the projection error. This ensures that the learned projection function maps the IMU features to a vector more similar to the video prototype vector.
- The second term $-\sum_{i=1}^{N} \sum_{k \in \mathcal{S}} y_i^k \cdot \log\left(p\left(y_i = c_k \mid \mathbf{z}_i\right)\right)$ denotes the cross-entropy loss on the prediction of the seen activity classes from the IMU vector.
- μ is a trade-off parameter between the projection and prediction losses.

5.2.5 Combining Semantic Spaces

In addition to the standard case of employing one semantic space, we may also employ multiple spaces for inference at test time, for instance by combining word and video spaces. In the following, we describe a method to combine two semantic spaces based on distance vector combinations.

First, we train two ZSL models using the two spaces separately. During inference, the test instance is passed through both models so that we would get two projections $\mathbf{z}^A$ and $\mathbf{z}^B$. Before nearest neighbor classification, in the respective semantic spaces of $\mathbf{z}^A$ and $\mathbf{z}^B$, we calculate their distances measurements to all label prototypes and get two distance vectors Δ^A and Δ^B. Then the combined distance vector for the space *(A+B)* is defined as:

$$\Delta^{comb} = (1 - \alpha)\Delta^A + \alpha\Delta^B, \tag{5.11}$$

where α is a parameter for adjusting which of the semantic spaces carries more weight in influencing the prediction. The combined distance vector Δ^{comb} is then used to identify the closest prototype and make a prediction.

5.3 Experimental Setup

This section describes the data collection conducted and gives implementation details about our empirical experiments.

5.3.1 IMU Data

Three publicly available datasets are used in our experiments, namely PAMAP2 [41] dataset, DaLiAc [43] and UTD-MHAD [58] datasets, which have been introduced in Section 2.1.1.2. In each dataset, we use data from tri-axial accelerometers and gyroscopes. We detail the pre-processing done for each dataset as follows

- **PAMAP2** Before feature extraction, we remove 10 seconds from the start and end of each activity recording to reduce the amount of noise. Following [41], we segment the raw data into sliding windows with a window size of 5.12 seconds and an overlap of 1 second. Within each window, we calculate the mean and standard deviation and then save them as features. In total, we extract 24222 instances, with each instance containing 36 dimensions.

- **DaLiAc** We merge two classes, *"bicycling on ergometer (50 W)"* and *"bicycling on ergometer (100 W)"*, into one class in order to ease the finding of corresponding videos for each activity class in later experiments. We again use the sliding window method with 5.12 seconds as window size with 1 second of overlap to extract mean and standard deviation features. This results in 21889 instances, with each instance containing 48 dimensions.

- **UTD-MHAD** UTD-MHAD provides time-synchronized RGB video and inertial data. Since each recording in UTD-MHAD is very short (< 5 seconds), we do not use the sliding window approach, but instead treat the whole recording as one

single window with one class label. This results in 861 instances, with each instance containing 12 dimensions.

Train-Test Split Following [101], we adopt a k-fold evaluation approach to split the activity classes in each dataset into train (seen) and test (unseen) portions over k disjoint folds. On PAMAP2, we adopt the same train-test split defined in [101] ($k = 5$), which leaves 3 to 4 unseen classes representing different activity types per fold. For DaLiAc, we randomly select 3 unseen classes in each of $k = 4$ folds. For UTD-MHAD, we randomly select 5 to 6 unseen classes in $k = 5$ folds. To ensure that the resulting train-test splits are balanced in the types of activities, in both cases we first manually identify the types of activities that a class belongs to, then randomly select classes from each type of activity to populate the test set of each fold. More details on train-test splits can be found in Appendix A.2.

5.3.2 Video Data

To construct the video embedding space, we must have at least one video clip which is used to extract the prototype vector for each (seen and unseen) activity class. In total, we collected 10 videos per class for the 18 activity classes in PAMAP2, and 13 activity classes in DaLiAc. For activity classes that appear in both datasets (e.g. *"walking"*), videos are reused for both datasets. Figure 5.2 demonstrates example frames of the collected videos for 6 random activities.

The data collection of videos for PAMAP2 and DaLiAc activities are guided by the activity class labels in these datasets. Where possible, we source annotated videos from public visual HAR datasets, namely HMDB51 [72], UCF101 [73] Kinetics [24, 25] and Real-World [50]; This is done by manually browsing videos annotated with semantically similar activities, e.g. we refer to videos annotated as *"using computer"* in Kinetics when sourcing videos for *"computer work"*. If not enough relevant videos cannot be found, we use keyword-driven search to download online videos from YouTube. As a result, the collected videos are a mixture of amateurly or professionally recorded, with varying visual quality, camera stability, backgrounds, identity and number of human subjects. There may also be a small number of frames in each clip where the said activity cannot be identified due to occlusion, lighting conditions, or blurriness. We cap the video duration at 2 minutes with a minimum of 5 seconds. To extract video embeddings from the I3D model, we randomly select 512 consecutive frames from each clip, which represents 17 seconds assuming an average frame rate of $50Hz$; shorter videos are repeated to the 512 frames before input.

For UTD-MHAD, we directly use the videos provided by the dataset to extract the embeddings. There are 32 video clips available for each class label, as 8 subjects performed each activity 4 times. These videos have limited variability as the camera and the background are always fixed, and the subjects are always located in the centre. The videos have a frame rate of $15Hz$ so we use 256 consecutive frames to extract video embeddings from the I3D model for consistency.

5.3.3 Baseline Semantic Spaces

Attribute Space We also consider attribute spaces, which act as baselines. These spaces are built with manually defined attributes. For the PAMAP2 dataset, we adopt the attributes from the work of Wang et al. [101], which is based on the movement of body (e.g. *"arms straight (or not)"*) and related objects & environment (e.g. *"indoor (or not)"*). We then manually define the attribute space for the DaLiAc dataset and the UTD-MHAD dataset using a similar set of attributes. We arrive at attribute spaces of PAMAP2, DaLiAc and UTD-MHAD with 42, 45 and 29 dimensions respectively. These are documented Appendix A.3.

Word Embeddings We consider two word representation models, Word2Vec [244] and GloVe [245] to extract word embeddings. Both models leverage the co-occurrence of words in a large text dataset to measure their similarity, thereby creating a word semantic space where words with similar meanings are closer, and vice versa. Although both Word2Vec and Glove embeddings are popular in zero-shot learning problems in other domains, Word2Vec embeddings were only recently introduced in the field of inertial activity recognition [238, 239], so our experimentation with Glove embeddings represent a novel application altogether.

To extract the word embeddings from class labels, we use the Word2Vec model trained on the Google News dataset, as well as the GloVe model trained on the Wikipedia 2014 and the Gigaword 5. Both models produce 300-dimensional word vectors. For class labels with multiple words such as *"car driving"*, we compute the average of the feature vector of every word. For the PAMAP2 and the DaLiAc dataset, we use class labels provided by the authors. We make slight alterations to the labels of the UTD-MHAD dataset to make them more consistent and descriptive (e.g. *"catch"* to *"hand catch"*).

Random Embedding Space We additionally consider a random embedding space as

Figure 5.2: Example frames from the collected videos belonging to activity classes in the PAMAP2 or DaLiAc datasets.

a sanity check for the effectiveness of the semantic spaces and the ZSL method. Here, each class prototype is a randomly generated 400-dimensional vector. Intuitively, a method using this random embedding space should have the accuracy of a random guess.

5.3.4 Model Implementations

Owing to our zero-shot learning task and k-fold cross-validation setup (Section 5.3.1), we keep the amount of hyperparameter tuning to a minimum so as not to be influenced by the unseen class result during model development. This is in keeping with best practices in the ZSL domain since tuning hyperparameters based on the cross-validation test results (which contains the unseen classes) would violate the zero-shot assumption [246, 247]. We therefore fix any required hyperparameters after reviewing the ranges of common settings in inertial activity recognition literature [31], and after a round of initial experiments which confirm that the relative performances of all methods are not changing. We arrive at the following: We set $\mu = 1$ following [238]. We employ the ADAM optimizer [186] with learning rate of 10^{-4}, adopted L_2 regularization with $\lambda = 10^{-4}$ and ran training with 15 epochs with a batch size of 64. Unless otherwise specified, we follow these configurations throughout the paper. The models and supporting functions are implemented in Python with the PyTorch [227] framework, and model training is carried out mainly on an NVIDIA Tesla P100 GPU or an NVIDIA GeForce RTX 2070 Max-Q GPU.

5.4 Experiments

In this section, we describe the experiments conducted to examine the proposed zero-shot learning pipeline, with a particular focus on the effectiveness of the video-based semantic space. In each experiment, we discuss the underlying question, methods used, the results along with any implications and analysis.

5.4.1 Effectiveness of the Video Semantic Space

The first question we investigate is if we are able to create a meaningful video semantic space to facilitate zero-shot learning in inertial activity recognition. *Does the Video Semantic Space Work?*

Method

We collect 10 videos per activity class to construct a video-based semantic space for zero-shot learning. We evaluate our pipeline on the three benchmark datasets, namely PAMAP2, DaLiAc, and UTD-MHAD, using the I3D-features space, the I3D-logits space as defined in Section 5.2.2, as well as their combined space I3D-both (following Section 5.2.5). For comparison, we also perform controlled tests on semantic spaces of random, attribute, and word embeddings. We use average per-class accuracy as our main evaluation metric throughout the paper as it is the most widely used metric for existing zero-shot learning methods [101]. We also report macro F1-Score here as it is a common metric used in inertial HAR evaluation.

Results

We report the performance comparison between video and other embeddings in Table 5.1. We see that video embeddings deliver strong ZSL performance when compared to alternate semantic spaces. Considering I3D-features and I3D-logits alone, they achieve consistently better performance than word embeddings by a large margin (accuracy improvement ranges from 13% to 33%). In addition, I3D-both is seen to give performance gains when compared to even the hand-crafted attribute space. Overall, this is a positive sign that videos of human activity, even those collected in-the-wild without much curation, encode a more informative representation for relating seen and unseen classes.

Interestingly, the difference in performance given by the two layers of the same video network (I3D-features and I3D-logits) is substantial (average difference of 12% in accuracy), and I3D-logits is seen to give better performance on PAMAP2 and UTD-MHAD. We believe this is related to the larger overlap in the nature of activities present in these datasets and Kinetics, on which the I3D model was pre-trained. Since the 400 dimensions of video-based class prototypes can be interpreted as a probability distribution over the 400 activity classes, this information may be more useful for representing the unseen classes found in PAMAP2 and UTD-MHAD (which, like Kinetics, contain many sports-related actions and activities). On the other hand, the I3D-features space extracts feature representations that are less fine-tuned to the particulars of the Kinetics label space, and may be more generally informative. The significant performance gains observed when combining both spaces into I3D-both seen on PAMAP2 and DaLiac (average gains of 8% in accuracy) indicate that these representations can be complementary and effectively combined to improve performance.

Visualizations

Figure 5.3 shows the per-activity F1 performances given by the semantic spaces of video, word and manual attributes when evaluated on the unseen classes in each fold. The results shown highlights that the zero-shot learning setting is indeed a challenging problem – since near-zero F1 scores are not uncommon in the considered tasks. Also, no single semantic space is able to outperform another consistently without fail, this applies even to the hand-crafted attribute space, which is seen to give completely wrong predictions for classes like *"watching TV"* in PAMAP2. The lack of consistent performance across activity classes suggest room for refinement in the current zero-shot learning setup, and it may not be suitable for robust industrial application yet.

The relative performances of semantic spaces in Figure 5.3 may reveal the limitations of each space. Word2Vec performs especially poorly in many activities which are predominantly characterized by their motions (such as *"running"* and *"playing soccer"* in PAMAP2 and *"descending stairs"* in PAMAP2 and DaLiAc), which may be related to the limited amount of motion-specific information contained in word-based spaces. Weak Word2Vec performance on some sedentary activities (such as *"watching TV"* in PAMAP2) may be attributed to the varying definitions of the composite word vectors, which may be too non-specific for activity identification. While the video-embedding space seems to perform well on most of the aforementioned activity examples, they still struggle in certain cases (e.g. confusing *"rope jumping"* with *"descending stairs"* or *"ascending stairs"*)

Semantic space		PAMAP2		DaLiAc		UTD-MHAD	
		Acc.	F1	Acc.	F1	Acc.	F1
Hand-crafted	Random	28.5	23.4	35.6	28.8	18.6	16.3
	Manual attributes	60.6	54.7	70.7	63.2	40.8	37.8
Learnt (Word)	Word2Vec	42.5	38.4	59.6	53.6	32.6	29.1
	GloVe	40.5	33.8	60.0	57.9	24.3	21.0
Learnt (Video)	I3D-features	49.3	45.7	65.5	**60.2**	36.4	32.8
	I3D-logits	**56.4**	**49.5**	**68.0**	58.0	**42.8**	**38.5**
	I3D-both	**65.5**	**61.2**	**71.6**	**66.2**	**43.4**	**39.5**

Table 5.1: Comparison of zero-shot learning approaches when different semantic spaces are used. Best results among the individual learned spaces are bolded. Best overall results are colored in blue.

where the motion characteristics alone may not lend enough specificity for their identification in a zero-shot learning setting.

The UTD-MHAD dataset presents the hardest scenario for all three types of semantic spaces, possibly because the activity classes here involve the most fine-grained distinctions between motions, e.g. *"swipe left"* was almost completely misclassified by all three spaces. We see word embeddings perform relatively well when class labels involve phrases that are prescriptive, such as *"pick up then throw"* and *"two hand push"* which neither video nor manual attributes spaces are able to recognize. We see poor Word2Vec performance in classes where the label is a single word, e.g. *"wave"* and *"squat"*, which suggests that common feature representations of verbs alone may not contain sufficient motion-related information needed for an IMU zero-shot learning problem. Moreover, the contrasting performance of word embeddings on *"sit then stand"* versus *"stand then sit"* suggests its difficulty in separating sequential actions; in contrast, both activities can still be classified to varying degrees by the video and attribute embedding space. In addition, we also see that video embeddings perform better in most cases where visual knowledge is expected (e.g. *"drawing circle clockwise"*).

5.4.2 Scalability of the Video Semantic Space

In the previous section, we curated 10 video clips per activity class in order to derive the video prototype vectors in PAMAP and DaLiAc. In this section, we examine how zero-shot classification performance varies with the amount of video data used. *Is the Video*

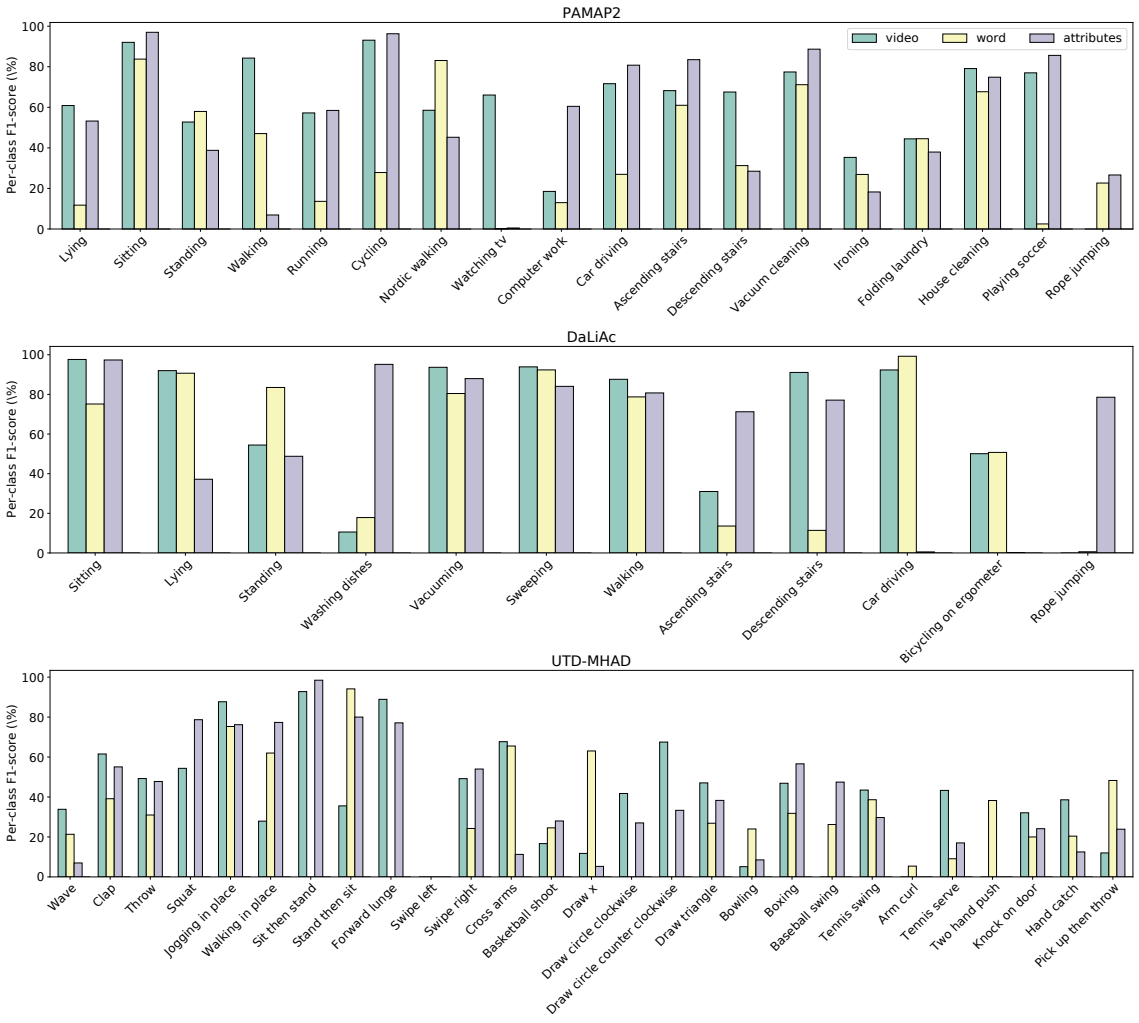


Figure 5.3: Per-class F1 scores of all activities encountered in k-fold evaluation on each dataset. ZSL approaches using I3D-both, the best-performing word-based space (i.e. GloVe on DaLiAc and Word2Vec on PAMAP2 and UTD-MHAD), and manual attributes are plotted here for comparison.

Semantic Space Scalable? We believe this is an important investigation that can shed light on the trade-offs between video data curation and model performance.

If we observe that the ZSL can perform reasonably well with fewer videos, this may translate to lower data curation costs as we need not curate 10 videos per activity. Additionally, we are also interested in seeing if the model performance increases with the number of videos added, as that would suggest that we can potentially make use of the vast amount of video resources to improve performance in inertial HAR and open up many exciting research opportunities.

Method

We conduct a new set of experiments on PAMAP2, DaLiAc, and UTD-MHAD by varying the number of videos per class n used to compute the video embedding vector. We randomly sample n videos from a total of $N = 10$ for PAMAP2 and DaLiAc, and $N = 32$ for UTD-MHAD. For each n, we repeat the sampling of videos 10 times and report the averaged results reached after training for 10 epochs per run. We conduct these experiments using the I3D-logits space.

Results

The experiment results are shown in Figure 5.4. A positive trend is observed in all three datasets, where classification accuracy is increased as the number of videos per class n is raised. The jump in accuracy is especially visible in the region of $n \leq 3$. Across the three datasets, one sees that the improvement gradually lowers as n gets closer to 10. This could indicate that 10 is already a good balance on the trade-off between accuracy and video collection, but if more accuracy is needed, there could still be some room for further growth. For the DaLiAc, it is even possible that the performance of the I3D features space could surpass that of the attribute space if after $n > 10$. In addition, it is worth noting that only $n \geq 3$ videos per class are enough for the video embedding spaces to outperform the word-based approaches.

We note that it is surprising to also observe such a positive trend on the UTD-MHAD dataset, since the 32 videos available per class (4 repeated actions by 8 subjects) have little visual difference when judging by the human eye. There is still an upward trend even when n is close to 32. Although we have adopted a simple averaging approach to aggregate multiple video embedding vectors into one class prototype, these results suggest that this simple strategy is enough to retain information from an increasing number of video data, such that the resulting increase in accuracy is still substantial.

Overall, we believe the results from this experiment reveal a desirable property for using video embeddings to provide auxiliary information for zero-shot learning, when compared to the attribute or word semantic space, which do not provide a similar route for further improvement. To increase the amounts of data, researchers can collect more videos in a similar manner to our data collection protocols described in Section 5.3.2, which have a much lower cost of curation than IMU data, or investigate well-established data augmentation strategies developed in computer vision (e.g. cropping, rotating). Therefore, researchers can make use of this property to carefully balance the trade-off in model performance and data curation cost according to their own needs.

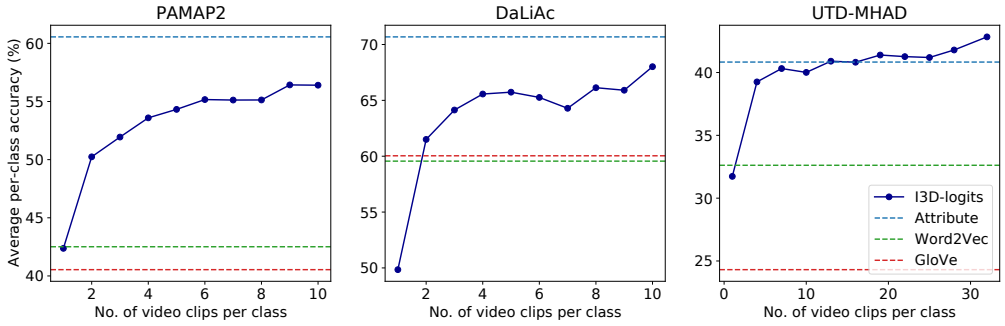


Figure 5.4: Per-class accuracy of our ZSL approach with I3D-logits as its semantic space is plotted for the three datasets, while the number of video clips per classed is increased from 1 to 10 on PAMAP2 and DaLiAc, and from 1 to 32 on UTD-MHAD. We observe an overall upward trend in model accuracy as more video data are used in constructing the video-based semantic space.

5.4.3 Combining Semantic Spaces

In Section 5.4.1, we saw that the word and video semantic space each offers a different approach to creating an informative class representation for zero-shot learning. Meanwhile, previous works in ZSL for general applications [248–251] and video-based HAR [252, 253] have found that performance improvements may be achieved by combining different semantic spaces; The intuition is that different semantic spaces encode complementary relationships between the classes, and such combination may also be seen as a variant of ensemble model learning. In this experiment, we investigate the combination of alternative perspectives from words and videos in constructing a joint semantic space. *Is it beneficial to combine learned embedding space for zero-shot learning problems in inertial HAR?*

Method

To answer this question, we construct combined word and video spaces using the trained models presented in Section 5.4.1 and report the best performance achieved by varying α from 0 to 1 in steps of 0.01 according to the formulation presented in Section 5.2.5.

Results

Table 5.2 reports the best ZSL performance achieved by the combined semantic spaces. Best performances are seen by combinations of I3D-logits with the best-performing word-based space on each dataset, which increases accuracy by 9% on both PAMAP2 and DaLiAc and by 5% on UTD-MHAD. Figure 5.5 shows how the performance accuracy changes as α is adjusted on each dataset, where a higher weighting for the video embedding is seen to be optimal for PAMAP2 and DaLiAc but not UTD-MHAD; this suggests

Semantic space		PAMAP2		DaLiAc		UTD-MHAD	
		Acc.	F1	Acc.	F1	Acc.	F1
Combined	I3D-features + Word2Vec	55.0	51.7	68.5	63.9	44.6	40.9
	I3D-features + GloVe	52.6	48.9	70.5	68.6	40.0	36.4
	I3D-logits + Word2Vec	**61.6**	**55.9**	68.9	64.4	**48.8**	**45.9**
	I3D-logits + GloVe	60.1	54.1	**74.0**	**70.2**	44.8	41.4

Table 5.2: Comparison of zero-shot performance when different word-based and video-based semantic spaces are combined. Best results are bolded.

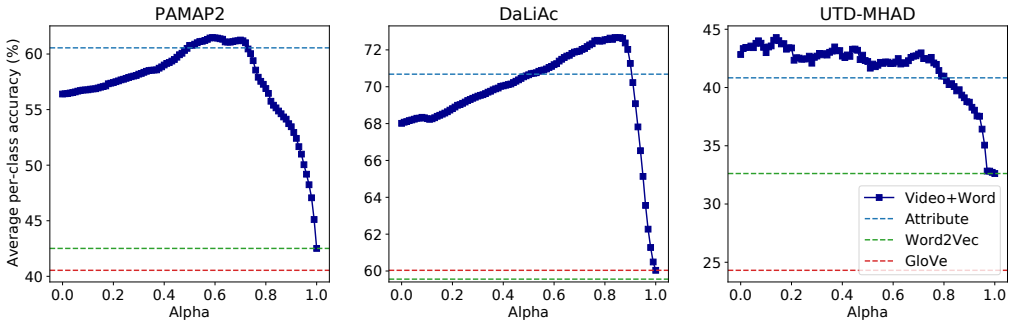


Figure 5.5: Classification accuracy when α is varied from 0 to 1 in steps of 0.01. $\alpha = 0$ represents the case of learning with video-based space only, and $\alpha = 1$ represents the case of word-based space only.

that the combination of the semantic spaces must be carefully considered for each setting.

The observed performance gains could be explained by a smoothing effect brought by combining the semantic spaces, which we illustrate using delta confusion matrices on example folds of the PAMAP2 dataset in Figure 5.6. Each delta confusion matrix is computed by subtracting the normalized confusion matrix of Word2Vec from that of I3D-logits or I3D-logits+Word2Vec and indicates their relative performances. We see the smoothing effect leading to synergies of the two spaces in cases where I3D-logits struggle, most noticeably in the confusion between *"Nordic walking"* and *"running"*. Here, using the video embedding space alone has led to unsatisfactory results, possibly due to the visual similarity between *"Nordic walking"* and *"running"*, whereas the word embeddings of these activities are clearly different, leading to benefits when considering both information together. However, we note that the current simple combination approach does not always improve performance, as the benefits of individual embedding spaces may also be smoothed out, e.g. identifying *"lying"* in PAMAP2. Inspection of delta matrices across the datasets led to similar findings. This observation, together with the varying accuracy seen in Figure 5.5,

both point to future work in investigating different techniques to fully exploit the benefits of combining semantic spaces.

5.5 Discussion

There are still challenges and obstacles that could constrain the practical application of video-embedding-based ZSL. In this section, we discuss the limitations and potential extensions of our work.

Activity Classes Found in IMU Datasets In order to evaluate ZSL systems, there needs to be a large number of activity classes so that the training and testing portions of the datasets would contain enough seen and unseen activities respectively. Evaluation of ZSL problems for IMU HAR is currently difficult due to the limited diversity of activity classes present in public datasets. In addition, since our method assumes the availability of video data corresponding to each activity class, the mismatch in activity class labels between IMU HAR datasets and Video HAR datasets also presents another limitation. As mentioned in Section 4.5, many IMU datasets have large amounts of data for static classes such as *"standing"*, *"sitting"*, *"lying"*, which are rarely captured in long durations in video dataset; Even when they are captured, the available video data often focus on the transitional portion of these activities (e.g. going from sitting to standing and vice versa) instead of the static activity itself.

Indistinguishable Videos There are a number of activities that may look visually similar but are actually different in reality and measurable by IMU data, e.g. *"lift going up"* and *"lift going down"*. We expect that video embedding space would fail on tasks seeking to distinguish such activities. We therefore suggest that a good ZSL pipeline should take this into consideration in the definition of activity classes and reflect activities that are both visually and inertially distinguishable; one might also consider combined semantic spaces so that the class representation does not only consider videos as its source.

Impact of Changing Video Quality We posit that our method is robust to common changes in video quality (e.g. blurriness, occlusion, varying lightness, changing camera angles) due to our use of pre-trained video activity recognition models which are robust to such variations. Specifically, we have employed the state-of-the-art I3D network, which

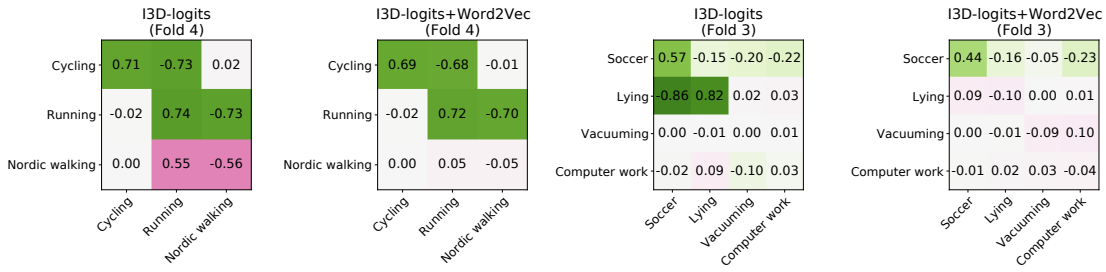


Figure 5.6: Delta confusion matrices between I3D/I3D+Word2Vec and the Word2Vec on Fold 4 and Fold 3 on the PAMAP2 dataset. The depicted folds are chosen because I3D-logits achieve the best and worst accuracies on them. From left to right: Best fold (I3D-logits), best fold (I3D-logits+Word2Vec), worst fold (I3D-logits), worst fold (I3D-logits+Word2Vec). In each matrix, better and worse performance of video embeddings over word embeddings is indicated in green and red respectively; Therefore, on the main diagonal, a positive number appears green, and elsewhere, a negative number appears green.

was pre-trained Kinetics-400, a dataset consisting of at least 400 clips per activity sourced from YouTube. Thus, the resulting semantic space is given by a model which has already encountered videos that vary greatly in quality (e.g. the camera framing, viewpoint, how the action was performed, clothing, body pose), importantly these include videos that have considerable camera motion/shake, illumination variations, shadows, background clutter, etc [24]. Together with image augmentation during training, we expect the model to be able to extract embeddings while allowing for common variations in video quality. We have qualitatively confirmed these assumptions by inspecting different subsets of randomly selected videos and their performances used in Figure 5.4, where we observed no meaningful correlations between video quality and ZSL results. As a systematic evaluation of the empirical impact of video quality on ZSL performances would require a larger curated video dataset, we leave this investigation as future work.

In the following, we highlight areas of potential extensions of this work.

Generalized Zero-Shot Learning None of the previous works in inertial HAR have attempted a generalized zero-shot learning (GZSL) setting [254], which generalizes the zero-shot learning task to one where both seen and unseen classes are classified at test time. This is a realistic scenario for activity recognition tasks, since the seen classes (which we have training IMU data for) are often the most frequently performed activities, so it is likely that the seen activities are also targeting activities of a classifier. Since our current model has only been trained with instances belonging to the seen classes, the resulting predictions will bias towards seen classes if our model was naively applied for GZSL. There are some universal GZSL algorithms such as two-stage methods [160, 255] in which we first classify

whether an instance belongs to seen classes or unseen classes, or the calibrated-stacking-based methods [256–258], in which the seen classes are deliberately disadvantaged during the final $argmax$, but these methods are not related to embedding spaces. We believe the more interesting and promising direction is the generative-based methods [259], in which a conditional generative model is trained to take the semantic prototype and synthesize an instance of the corresponding class, and then many instances of unseen classes could be generated to train a classifier. We anticipate that if the semantic prototype is from a video semantic space, it should contain much richer domain-specific information, which could lead to more realistic synthesized instances.

Data Augmentation on Videos As shown in Section 5.4.2, we observe that the classification accuracy increases with the number of videos used in constructing the semantic space, even when the videos are visually similar to each other in the case of UTD-MHAD. While collecting or recording more videos may incur additional time, it can be flexibly done depending on the research resources available. Another way to artificially generate more video instances with minimum cost is to perform data augmentation on the base videos. Spatially, random cropping, flipping and rotation could be easily applied to video frames. Temporally, window slicing may also be used to break long videos into shorter pieces.

Skeleton-Based HAR Models To further reduce the domain gap between the IMU features and the class prototypes generated from video data, we may consider models trained using other forms of visual information. In Chapter 4, we have seen that a promising class of models are found in skeleton-based activity recognition [260]. The intuition is that the key points in the skeleton explicitly represents the movement of body parts, and activity recognition models from these skeleton joints will only focus on the video's motion-related information, just like inertial HAR models. Many open-sourced toolboxes exist which allow the easy extraction of skeleton joints from RGB videos (e.g. OpenPose [109]). In addition, if researchers are considering recording their own videos, using skeleton-based data are better for preserving the identities of human subjects.

5.6 Related Work

Few previous works which implement zero-shot learning exist for the application of IMU-based activity recognition. Cheng et al. [237] proposed a method that is similar to DAP

[261], which employs separate SVMs to predict each attribute of a binary attribute semantic space. Thereafter, a typical nearest neighbour classifier is used for classification. Cheng et al. [262] later extended their work by replacing the SVMs with the conditional random field (CRF) and the nearest neighbour classifier with the junction tree algorithm [263], but the attribute semantic space remains the same. Wang et al. [101] proposed a nonlinear-compatibility-based method, which is equivalent to using an MLP to project from the feature space to the attribute semantic space. Ohashi et al.'s work [264] employs a CNN to extract features from raw IMU data and perform the projection at the same time. Their semantic space is still an attribute space, but it contains binary, discrete and continuous attributes. Moreover, they also introduced the idea of attributes' importance so that instances from different classes would have different emphasis on the attributes, but the importance table is also manually defined. The use of learnt semantic spaces, as defined by word embeddings, was not studied until recent works [238, 239]. In [239], Matsuki et al. compared the use of attribute vectors and two variations of word embedding vectors and found their performances to be similar. Most recently, Wu et al. [238] proposed an MLP with skip connections for projection, and they found in their experiments that a manually defined attribute space gives superior performance to a semantic space defined by Word2Vec [244]. While previous work predominantly utilises the manual attribute space, and even word embedding spaces have not been fully explored, our work instead explores both word embedding spaces and the novel video embedding space.

5.7 Summary

In this chapter, we propose a strategy to exploit videos of human activities to construct an informative semantic space for zero-shot learning problems on inertial activity recognition. This novel video semantic space effectively facilitates the knowledge transfer from state-of-the-art video action recognition models to unseen activity recognition using IMU data. We evaluated our method on three public IMU datasets for zero-shot learning and compared the results against traditional learned and hand-crafted embedding spaces. Our results show that our video embedding space consistently outperforms a word embedding space, and is comparable to the accuracy of the manual attribute space. With these encouraging results, our investigation suggests that exploiting videos of human activities as well

as their associated models developed in the computer vision space has great potential for improving activity recognition tasks using IMU data.

Chapter 6

Conclusion

We review the research questions posed at the beginning of this thesis, and summarize the contributions made to address these questions. Following this, we discuss future directions highlighted by our work in the field of inertial Human Activity Recognition.

6.1 Summary

The primary goal of this thesis is to promote an approach for activity recognition on wearable inertial sensors that is not impeded by data collection costs – by exploiting the visual domain. This thesis has studied this challenging problem from three different perspectives: First, by identifying a new paradigm for virtual IMU data generation from videos of human activities; Second, by facilitating the knowledge transfer from activity recognition models trained using visual information to those operating on inertial data; Third, by designing the use of video embeddings to enable zero-shot classification of activities that were never previously seen during training.

We revisit our research questions and provide a summary of our major contributions below:

Question 1 *Can we design a pipeline that, upon input of a video depicting human activities, generates a virtual measurement time series from inertial sensors placed on the human body? To what extent can we replace the real sensor data with the virtually generated sensor data for training activity recognition models? What learning frameworks will allow us*

to draw benefits from the virtual data for activity recognition?

In Chapter 3, we studied algorithms for converting video data to virtual IMU data, which are then used to train HAR models that classify real IMU data at test time. With our proposed pipeline, IMUTube, we extracted virtual IMU data from raw videos of human activities, offering a novel route for virtually generating IMU data to be used for building inertial HAR models. Through extended evaluation of IMUTube we show that, by exploiting only a small amount of real IMU data for calibration purposes, activity recognition models trained with its generated virtual IMU data achieve up to 98% of the performance given by models trained only with real data. We additionally demonstrate two opportunities in which virtual data may be used alongside real data to improve HAR model learning: First, models trained with a naive mixture of virtual and real data provide up to 11% performance gains over those trained with real data only; Second, deep models pre-trained with labeled virtual data provide up to 14% performance increase over models trained from random initializations. These results demonstrate the promising potential of using IMUTube to generate virtual IMU data that is helpful for building activity recognition models. While we also observe that our current proof-of-concept pipeline is not yet suited to yielding benefits for complex activity recognition, we carry out additional investigations to better understand the capabilities and limitations of IMUTube and provide feasible extensions for future work to overcome them.

Question 2 *Can the knowledge encoded in HAR models from the vision domain be exploited to learn enriched feature representations and improve activity recognition performance in the inertial domain?*

In Chapter 4 we studied the transfer of knowledge from an existing visually-trained activity recognition model to train a model operating on IMU data. Our examination of the cross-modal knowledge transfer problem revealed challenges in a straightforward application of existing techniques, with a major challenge being the lack of large-scale paired visual and inertial data. We overcame this barrier by developing a solution that is equipped to face the unique challenges in visual-to-inertial knowledge transfer – without requiring paired video and IMU data. Our method is able to transfer knowledge learned from a state-of-the-art pose-based activity recognition network to a network that can only extract that information from the IMU counterpart; the benefits of knowledge transfer are seen empirically in terms of performance gains, as well as signs of visual grounding in the learned representations of inertial data.

Question 3 *How do different auxiliary embedding spaces compare in zero-shot activity recognition performance? Can we construct a video-based embedding space to enable zero-shot inference on on-body inertial data?*

In Chapter 5, we studied the zero-shot learning problem in inertial activity recognition using three auxiliary semantic spaces: hand-crafted attributes, word embeddings, and video embeddings. To the best of our knowledge, this is the first work that incorporates video embeddings for ZSL applications both within and outside the field of inertial HAR. We design the construction of a simple but informative semantic space, using videos of human activities. With our approach, knowledge from state-of-the-art video activity recognition models is encoded into video embeddings to relate seen and unseen activity classes. Our experiments demonstrated that video embeddings outperform other learned (word) embeddings, and may be combined to give even better ZSL performance. Moreover, we learned that our method of constructing video-based semantic spaces can offer an additional desirable feature of scalability, as recognition performance was seen to scale with the amount of video data used.

6.2 Future Work

The work carried out in this thesis shed light on some of the most immediate questions inspired by cross-modal thinking. Here we discuss some of the most promising future directions that our work can take to extend their contributions.

Integrating modalities beyond vision and IMU While the cross-modal ideas presented in this thesis have been limited to knowledge transfer from the visual to the inertial domain, it is reasonable to conceive these ideas extending their applications to other modalities. The abundance of data and machine learning models developed for language and audio understanding suggests that these may be promising candidates as source modalities – we have already seen the potential benefits of using both visual and word-based embeddings for zero-shot learning in Chapter 5, and the existence of multi-modal datasets (e.g. [265]) may facilitate further investigation in this direction. On the other hand, many new sensing modalities that are emerging in our ubiquitous environments (e.g. physiological sensors [266]) may be seen as target modalities for cross-modal learning, since there will be many shared challenges between exploiting inertial sensors and these upcoming sensors for activity recognition on wearable devices. While modality-specific solutions will need to be developed, we are confident that the cross-modal learning methods studied in this thesis will

remain as a useful reference; indeed we have already seen a spillover of our video-based virtual data generation idea to adjacent domains, as seen in vid2Doppler, a framework that generates virtual Doppler radar data from videos for activity recognition [267].

Cross-modal self-supervision Concurrent to the investigations of cross-modal learning approaches in this thesis, self-supervised learning (SSL) techniques have emerged as a promising approach in inertial HAR [225, 268–270]. Self-supervised learning aims to exploit large amounts of unlabeled IMU data to learn better models, and the general approach is to pre-train a model using "pretext" tasks, followed by fine-tuning on a small labeled dataset. However, pretext tasks that have been explored so far in inertial HAR have been limited to a low-level understanding of the time series (e.g. masked reconstruction [269], rotation [225]), and thus may not have maximized their potential for the high-level activity understanding required in inertial HAR. We believe that pretext tasks that are more tailored to this complex downstream task may be defined by taking a cross-modal view that relates unlabeled activity data from both vision and IMU modalities. Indeed, cross-modal self-supervision has been successfully applied for audio and vision through unlabeled videos which synchronize the modalities [271]; while it is unclear whether such an approach can translate its benefits to unpaired visual and inertial data, we believe that including a cross-modal alignment step will be a promising strategy for self-supervision on unpaired visual and IMU data (a recent example implementing this idea can already be found in [272]), which can be possibly extended to other modalities and unpaired scenarios.

Multi-modal datasets Most inertial HAR datasets only provide data from the collected inertial sensor data and the activity class annotations, without any other modalities (e.g. video or audio recordings) which can be interpreted retrospectively by other researchers to gather more information about the activities performed; This restricts the development of human activity recognition systems to consider only a classification task – one that was defined by the researchers collecting the data, and poses great challenges in investigation of potential annotation errors and dataset artifacts [273]. We believe that an increasing availability of multi-modal datasets describing human activities (examples can be found in [265, 274]) will allow an extended investigations of cross-modal learning methods introduced in this thesis. Moreover, we believe a promising opportunity presented by multi-modal datasets will be to introduce new tasks that facilitate different aspects of automated activity understanding, for example, a *cross-modal retrieval* task [275] where the model should retrieve the most closely-related video clip give an IMU sample. Defining such cross-modal tasks may be a good way for model evaluation and interpretation, since we can now see IMU features as being related to visually-interpretable forms of data that we

as humans can understand. One of the most common challenge present throughout this thesis is the lack of paired data between the visual and inertial modality, which is true in the public and academic domain. We believe that this thesis may hold interesting implications to technology companies, where innovative ways of generating large amounts of passively collected, yet paired, multi-modal activity data may be explored. A case in point is smartwatch wearers who record or follow online fitness videos, a popular form of physical workout conducted via social media platforms [276]; Paired multi-modal data may be found in the video data that captures the fitness instructor's movements, together with the inertial data of the smartwatch user watching the video (and potentially also from the fitness instructor's smartwatch).

References

[1] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag Pedja, Karl Koscher, Anthony LaMarca, et al. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2):32–41, 2008.

[2] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *IEEE Communications magazine*, 48(9):140–150, 2010.

[3] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *23th International conference on architecture of computing systems 2010*, pages 1–10. VDE, 2010.

[4] Godwin Ogbuabor and Robert La. Human activity recognition for healthcare using smartphones. In *Proceedings of the 2018 10th international conference on machine learning and computing*, pages 41–46, 2018.

[5] Abdulhamit Subasi, Mariam Radhwan, Rabea Kurdi, and Kholoud Khateeb. Iot based mobile healthcare system for human activity recognition. In *2018 15th learning and technology conference (L&T)*, pages 29–34. IEEE, 2018.

[6] Ignacio Perez-Pozuelo, Dimitris Spathis, Emma AD Clifton, and Cecilia Mascolo. Wearables, smartphones, and artificial intelligence for digital phenotyping and health. In *Digital Health*, pages 33–54. Elsevier, 2021.

[7] Gabriella M Harari, Nicholas D Lane, Rui Wang, Benjamin S Crosier, Andrew T Campbell, and Samuel D Gosling. Using smartphones to collect behavioral data in psychological science: Opportunities, practical considerations, and challenges. *Perspectives on Psychological Science*, 11(6):838–854, 2016.

[8] Gabriella M Harari, Sumer S Vaid, Sandrine R Müller, Clemens Stachl, Zachariah Marrero, Ramona Schoedel, Markus Bühner, and Samuel D Gosling. Personality sensing for theory development and assessment in the digital age. *European Journal of Personality*, 34(5):649–669, 2020.

[9] Valentina Camomilla, Elena Bergamini, Silvia Fantozzi, and Giuseppe Vannozzi. Trends supporting the in-field use of wearable inertial sensors for sport performance evaluation: A systematic review. *Sensors*, 18(3):873, 2018.

[10] Joyce Ho and Stephen S Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 909–918, 2005.

[11] Jeffrey W Lockhart, Tony Pulickal, and Gary M Weiss. Applications of mobile activity recognition. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 1054–1058, 2012.

[12] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 3–14, 2014.

[13] Nils Yannick Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz. Pd disease state assessment in naturalistic environments using deep learning. In *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[14] Agnes Grünerbl, Amir Muaremi, Venet Osmani, Gernot Bahle, Stefan Oehler, Gerhard Tröster, Oscar Mayora, Christian Haring, and Paul Lukowicz. Smartphone-based recognition of states and state changes in bipolar disorder patients. *IEEE journal of biomedical and health informatics*, 19(1):140–148, 2014.

[15] Kieran Woodward, Eiman Kanjo, Muhammad Umair, and Corina Sas. Harnessing digital phenotyping to deliver real-time interventional bio-feedback. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 1206–1209, 2019.

[16] Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. How busy are you? predicting the interruptibility intensity of mobile users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5346–5360, 2017.

[17] Fabio Buttussi and Luca Chittaro. Mopet: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine*, 42(2):153–163, 2008.

[18] Kenneth R Foster and John Torous. The opportunity and obstacles for smartwatches and wearable sensors. *IEEE pulse*, 10(1):22–25, 2019.

[19] Aiden R Doherty, Niamh Caprani, Ciarán Ó Conaire, Vaiva Kalnikaite, Cathal Gurrin, Alan F Smeaton, and Noel E O'Connor. Passively recognising human activities through lifelogging. *Computers in human behavior*, 27(5):1948–1958, 2011.

[20] Catherine Tong, Shyam A Tailor, and Nicholas D Lane. Are accelerometers for activity recognition a dead-end? In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, pages 39–44, 2020.

[21] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.

[22] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, 2014.

[23] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[24] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[25] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *CoRR*, abs/1808.01340, 2018.

[26] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *CoRR*, abs/2010.10864, 2020.

[27] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. Imutube: 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):1–29, 2020.

[28] Catherine Tong and Nicholas D Lane. Empowering imu-based activity recognition through cross-modal knowledge transfer from unpaired videos. *Work in Submission*, 2022.

[29] Catherine Tong, Jinchen Ge, and Nicholas D Lane. Zero-shot learning for imu-based activity recognition using video embeddings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–23, 2021.

[30] VW-S Tseng, Sourav Bhattachara, Javier Fernández-Marqués, Milad Alizadeh, Catherine Tong, and Nicholas D Lane. Deterministic binary filters for convolutional neural networks. International Joint Conferences on Artificial Intelligence Organization, 2018.

[31] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. Multimodal deep learning for activity and context recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–27, 2018.

[32] Catherine Tong, Matthew Craner, Matthieu Vegreville, and Nicholas D Lane. Tracking fatigue and health state in multiple sclerosis patients using connnected wellness devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–19, 2019.

[33] Christian Schroeder de Witt, Catherine Tong, Valentina Zantedeschi, Daniele De Martini, Alfredo Kalaitzis, Matthew Chantry, Duncan Watson-Parris, and Piotr Bilinski. Rainbench: Towards data-driven global precipitation forecasting from satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14902–14910, 2021.

[34] Catherine Tong, Emma Rocheteau, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting patient outcomes with graph representation learning. In *International Workshop on Health Intelligence*, pages 281–293. Springer, 2021.

[35] Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *arXiv preprint arXiv:2206.02909*, 2022.

[36] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, 22(4):1476, 2022.

[37] D. Titterton, J.L. Weston, J. Weston, Institution of Electrical Engineers, American Institute of Aeronautics, and Astronautics. *Strapdown Inertial Navigation Technology*. IEE Radar Series. Institution of Engineering and Technology, 2004.

[38] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[39] Thomas PlÖtz. Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them. *ACM Computing Surveys (CSUR)*, 54(6):1–25, 2021.

[40] R. Chavarriaga, H. Sagha, and D. Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letter*, 34(15):2033–2042, 2013.

[41] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, New York, NY, USA, 2012. Association for Computing Machinery.

[42] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. pages 127–140, 2015.

[43] Heike Leutheuser, Dominik Schuldhaus, and Bjoern M. Eskofier. Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset. *PLOS ONE*, 8(10):1–11, 10 2013.

[44] Rebecca Adaimi and Edison Thomaz. Leveraging active learning and conditional mutual information to minimize data annotation in human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–23, 2019.

[45] Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, and Nadir Weibel. Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[46] G. Laput and C. Harrison. Sensing fine-grained hand activity with smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[47] Wenqiang Chen, Shupei Lin, Elizabeth Thompson, and John Stankovic. Sensecollect: We need efficient ways to collect on-body sensor-based human activity data! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–27, 2021.

[48] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[49] Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. On-body device localization for health and medical monitoring applications. In *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 37–44. IEEE, 2011.

[50] Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2016.

[51] Gary M Weiss, Kenichi Yoneda, and Thaier Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

[52] Ian Cleland, Basel Kikhia, Chris Nugent, Andrey Boytsov, Josef Hallberg, Kåre Synnes, Sally McClean, and Dewar Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7):9183–9200, 2013.

[53] Thomas Plotz, Chen Chen, Nils Y Hammerla, and Gregory D Abowd. Automatic synchronization of wearable sensors and video-cameras for ground truth annotation–a practical approach. In *2012 16th international symposium on wearable computers*, pages 100–103. IEEE, 2012.

[54] Matthew Willetts, Sven Hollowell, Louis Aslett, Chris Holmes, and Aiden Doherty. Statistical machine learning of sleep and physical activity phenotypes from sensor data in 96,220 uk biobank participants. *Scientific reports*, 8(1):1–10, 2018.

[55] Tudor Miu, Paolo Missier, and Thomas Plötz. Bootstrapping personalised human activity recognition models using online active learning. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 1138–1147. IEEE, 2015.

[56] Kundan Krishna, Deepali Jain, Sanket V Mehta, and Sunav Choudhary. An lstm based system for prediction of human activities with durations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–31, 2018.

[57] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9, 2008.

[58] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 168–172, 2015.

[59] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*, pages 1–17. Springer, 2004.

[60] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. Collossl: Collaborative self-supervised learning for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–28, 2022.

[61] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar. A systematic study of unsupervised domain adaptation for robust human-activity recognition. 4(1), March 2020.

[62] Y. Guan and T. Plötz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)*, 1(2):1–28, 2017.

[63] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*, pages 1–6, 2018.

[64] Eduardo Velloso, Andreas Bulling, Hans Gellersen, Wallace Ugulino, and Hugo Fuks. Qualitative activity recognition of weight lifting exercises. In *Proceedings of the 4th Augmented Human International Conference*, pages 116–123, 2013.

[65] Dan Morris, T Scott Saponas, Andrew Guillory, and Ilya Kelner. Recofit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3225–3234, 2014.

[66] David Strömbäck, Sangxia Huang, and Valentin Radu. Mm-fit: Multimodal deep learning for automatic exercise logging across sensing devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–22, 2020.

[67] Cassim Ladha, Nils Y Hammerla, Patrick Olivier, and Thomas Plötz. Climbax: skill assessment for climbing enthusiasts. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 235–244, 2013.

[68] E. Thomaz, I. Essa, and G. Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1029–1040, 2015.

[69] George Vavoulas, Matthew Pediaditis, Emmanouil G Spanakis, and Manolis Tsiknakis. The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones. In *13th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–4. IEEE, 2013.

[70] Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, and Gerhard Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2):42–50, 2008.

[71] Naoya Yoshimura, Jaime Morales, and Takuya Maekawa. OpenPack: Public multi-modal dataset for packaging work recognition in logistics domain. March 2022.

[72] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[73] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[74] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[75] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *CoRR*, abs/1604.01753, 2016.

[76] C. Gu, C. Sun, D. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056, 2018.

[77] Yuya Yoshikawa, Jiaqing Lin, and Akikazu Takeuchi. Stair actions: A video dataset of everyday home actions. *arXiv preprint arXiv:1804.04326*, 2018.

[78] Nils Y Hammerla, Reuben Kirkham, Peter Andras, and Thomas Ploetz. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 international symposium on wearable computers*, pages 65–68, 2013.

[79] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[80] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[81] Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christoper G Owen, et al. Large scale population assessment of physical activity using wrist worn accelerometers: the uk biobank study. *PloS one*, 12(2):e0169649, 2017.

[82] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[83] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, volume 5. Citeseer, 2005.

[84] Katherine Ellis, Jacqueline Kerr, Suneeta Godbole, and Gert Lanckriet. Multi-sensor physical activity recognition in free-living. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 431–440, New York, NY, USA, 2014. Association for Computing Machinery.

[85] Thomas Plötz, Nils Y Hammerla, and Patrick L Olivier. Feature learning for activity recognition in ubiquitous computing. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[86] F. J. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[87] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015.

[88] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[89] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 197–205, 2014.

[90] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[91] Sojeong Ha and Seungjin Choi. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *2016 international joint conference on neural networks (IJCNN)*, pages 381–388. IEEE, 2016.

[92] Sojeong Ha, Jeong-Min Yun, and Seungjin Choi. Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International conference on systems, man, and cybernetics*, pages 3017–3022. IEEE, 2015.

[93] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 ieee international conference on big data and smart computing (bigcomp)*, pages 131–134. IEEE, 2017.

[94] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

[95] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. Lstm networks for mobile human activity recognition. In *2016 International conference on artificial intelligence: technologies and applications*, pages 50–53. Atlantis Press, 2016.

[96] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*, pages 351–360, 2017.

[97] Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. Innohar: A deep neural network for complex human activity recognition. *Ieee Access*, 7:9893–9902, 2019.

[98] Yuta Yuki, Junto Nozaki, Kei Hiroi, Katsuhiko Kaji, and Nobuo Kawaguchi. Activity recognition using dual-convlstm extracting local and global features for shl recognition challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, page 1643–1651, New York, NY, USA, 2018. Association for Computing Machinery.

[99] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven. Improving deep learning for har with shallow lstms. In *2021 International Symposium on Wearable Computers*, pages 7–12, 2021.

[100] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

[101] Wei Wang, Chunyan Miao, and Shuji Hao. Zero-shot human activity recognition via nonlinear compatibility based method. page 322–330, 2017.

[102] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[103] Artur Jordao, Antonio C Nazare Jr, Jessica Sena, and William Robson Schwartz. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. *arXiv preprint arXiv:1806.05226*, 2018.

[104] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1112–1123, 2016.

[105] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In *IJCAI*, pages 3109–3115, 2019.

[106] Hyunsung Cho, Akhil Mathur, and Fahim Kawsar. Flame: Federated learning across multi-device environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(3), sep 2022.

[107] Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[108] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019.

[109] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018.

[110] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

[111] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: 2000-speaker neural text-to-speech. In *International Conference on Learning Representations*, 2018.

[112] Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE, 2019.

[113] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[114] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[115] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[116] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020.

[117] Agisilaos Chartsias, Thomas Joyce, Rohan Dharmakumar, and Sotirios A Tsaftaris. Adversarial image synthesis for unpaired multi-modal cardiac data. In *International workshop on simulation and synthesis in medical imaging*, pages 3–13. Springer, 2017.

[118] Jelmer M Wolterink, Anna M Dinkla, Mark HF Savenije, Peter R Seevinck, Cornelis AT van den Berg, and Ivana Išgum. Deep mr to ct synthesis using unpaired data. In *International workshop on simulation and synthesis in medical imaging*, pages 14–23. Springer, 2017.

[119] Zizhao Zhang, Lin Yang, and Yefeng Zheng. Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 9242–9251, 2018.

[120] Yuta Hiasa, Yoshito Otake, Masaki Takao, Takumi Matsuoka, Kazuma Takashima, Aaron Carass, Jerry L Prince, Nobuhiko Sugano, and Yoshinobu Sato. Cross-modality image synthesis from unpaired data using cyclegan. In *International workshop on simulation and synthesis in medical imaging*, pages 31–41. Springer, 2018.

[121] AC Yew. Numerical differentiation: finite differences. *Brown University: Providence, RI, USA*, 2011.

[122] A. Young, M. Ling, and D. Arvind. Imusim: A simulation environment for inertial sensing algorithm design and evaluation. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, pages 199–210. IEEE, 2011.

[123] P. Asare, R. Dickerson, X. Wu, J. Lach, and J. Stankovic. Bodysim: A multi-domain modeling and simulation framework for body sensor networks research and design. In *International Conference on Body Area Networks (BODYNETS)*. ICST, 10 2013.

[124] Y. Huang, M. Kaufmann, E. Aksan, M. Black, O. Hilliges, and G. Pons-Moll. Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

[125] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, volume 36, pages 349–360. Wiley Online Library, 2017.

[126] S. Takeda, T. Okita, P. Lago, and S. Inoue. A multi-sensor setting activity recognition simulation tool. In *Proceedings of the ACM International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1444–1448, 2018.

[127] Carnegie Mellon Graphics Lab. *Carnegie Mellon Motion Capture Database*, 2008.

[128] N. Mahmood, N. Ghorbani, N. Troje, G. Pons-Moll, and M. Black. Amass: Archive of motion capture as surface shapes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5442–5451, 2019.

[129] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li. A deep learning method for complex human activity recognition using virtual wearable sensors. *arXiv preprint arXiv:2003.01874*, 2020.

[130] *Unity*.

[131] C. Kang, H. Jung, and Y. Lee. Towards machine learning with zero real-world data. In *The ACM Workshop on Wearable Systems and Applications*, pages 41–46, 2019.

[132] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[133] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[134] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[135] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[136] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

[137] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 826–834, 2016.

[138] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[139] Samuel Albanie, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Emotion recognition in speech using cross-modal transfer in the wild. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 292–301, 2018.

[140] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.

[141] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932*, 2017.

[142] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010.

[143] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[144] Fida Mohammad Thoker and Juergen Gall. Cross-modal knowledge distillation for action recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 6–10. IEEE, 2019.

[145] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[146] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[147] Chiho Choi, Sangpil Kim, and Karthik Ramani. Learning hand articulations by hallucinating heat distribution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3104–3113, 2017.

[148] José Lezama, Qiang Qiu, and Guillermo Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6628–6637, 2017.

[149] Andres Perez, Valentina Sanguineti, Pietro Morerio, and Vittorio Murino. Audio-visual model distillation using acoustic images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2854–2863, 2020.

[150] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Chris Xiaoxuan Lu, Yasin Almalioglu, Stefano Rosa, Changhao Chen, Johan Wahlström, Wei Wang, Andrew Markham, and Niki Trigoni. Deeptio: A deep thermal-inertial odometry with visual hallucination. *IEEE Robotics and Automation Letters*, 5(2):1672–1679, 2020.

[151] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[152] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021.

[153] Kang Li, Lequan Yu, Shujun Wang, and Pheng-Ann Heng. Towards cross-modality medical image segmentation with online mutual knowledge distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 775–783, 2020.

[154] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[155] Qi Dou, Quande Liu, Pheng Ann Heng, and Ben Glocker. Unpaired multi-modal segmentation via knowledge distillation. *IEEE transactions on medical imaging*, 39(7):2415–2425, 2020.

[156] Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019.

[157] Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414, 2014.

[158] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. *Advances in neural information processing systems*, 22, 2009.

[159] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[160] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 935–943, Red Hook, NY, USA, 2013. Curran Associates Inc.

[161] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009.

[162] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR 2011*, pages 3337–3344. IEEE, 2011.

[163] Donghui Wang, Yanan Li, Yuetan Lin, and Yueting Zhuang. Relational knowledge transfer for zero-shot learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[164] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016.

[165] Qian Wang and Ke Chen. Alternative semantic representations for zero-shot human action recognition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 87–102. Springer, 2017.

[166] Barbara Di Ventura, Caroline Lemerle, Konstantinos Michalodimitrakis, and Luis Serrano. From in vivo to in silico biology and back. *Nature*, 443(7111):527–533, 2006.

[167] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings of the ACM International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.

[168] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[169] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[170] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[171] I. Joel, A.and Stergios. A direct least-squares (DLS) method for PnP. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, November 2011.

[172] S. Shah and J.K. Aggarwal. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29(11):1775–1788, November 1996.

[173] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, March 1990.

[174] O. Bogdan, V. Eckstein, F. Rameau, and J. Bazin. Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the ACM SIGGRAPH European Conference on Visual Media Production*, pages 6:1–6:10. ACM, December 2018.

[175] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2004.

[176] H. Zhuang. A self-calibration approach to extrinsic parameter estimation of stereo cameras. *Robotics and Autonomous Systems*, 15(3):189–197, August 1995.

[177] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001.

[178] P.J. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[179] F. Pomerleau, F. Colas, and R. Siegwart. A review of point cloud registration algorithms for mobile robotics. *Found. Trends Robot*, 4(1):1–104, May 2015.

[180] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[181] J. Park, Q. Zhou, and V. Koltun. Colored point cloud registration revisited. In *IEEE International Conference on Computer Vision (ICCV)*, pages 143–152, 2017.

[182] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[183] T. Pham and Y. Suh. Spline function simulation data generation for walking motion using foot-mounted inertial sensors. In *Sensors*, pages 199–210. MDPI, 2018.

[184] P. Karlsson, B. Lo, and G. Z. Yang. Inertial sensing simulations using modified motion capture data. In *Proceedings of the International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 16–19, 2014.

[185] W. Conover and R. Iman. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician*, 35(3):124–129, 1981.

[186] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[187] T. Um, F. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the ACM International Conference on Multimodal Interaction*, pages 216–220, 2017.

[188] N. Y. Hammerla, S. Halloran, and T. Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1533–1540. AAAI Press, July 2016.

[189] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[190] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[191] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2556–2563. IEEE, 2011.

[192] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[193] K. Soomro, A. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[194] G. Sigurdsson, G. Varol, X. Wang, I. Laptev, A. Farhadi, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *arXiv preprint arXiv:1604.01753*, 2016.

[195] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–14, 2010.

[196] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Duan, and A. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[197] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *British Machine Vision Conference (BMVC)*, 2017.

[198] N. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 355–364. ACM, 2011.

[199] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709, 2018.

[200] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[201] Harish Haresamudram, David V Anderson, and Thomas Plötz. On the role of features in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*, pages 78–88, 2019.

[202] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. Cross-people mobile-phone based activity recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[203] A. Reiss and D. Stricker. Personalized mobile physical activity recognition. In *Proceedings of the ACM International Symposium on Wearable Computers*, pages 25–28, 2013.

[204] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[205] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[206] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the international conference on machine learning (ICML)*, pages 807–814, 2010.

[207] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2965–2972, 2014.

[208] S. Alireza Golestaneh and L. Karam. Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5800–5809, 2017.

[209] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[210] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[211] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5572–5581, 2019.

[212] J. Yu and R. Ramamoorthi. Robust video stabilization by optimization in cnn weight space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3800–3808, 2019.

[213] T. Zhou, M. Brown, Noah S., and D. Lowe. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017.

[214] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12882–12891, June 2022.

[215] G. Pons-Moll, J. Romero, N. Mahmood, and M Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015.

[216] A. Kanazawa, M. Black, D. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018.

[217] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):1–16, 2015.

[218] M. Rosca, B. Lakshminarayanan, and S. Mohamed. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

[219] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[220] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60. IEEE, 2013.

[221] V. Rey, P. Hevesi, O. Kovalenko, and P. Lukowicz. Let there be imu data: generating training data for wearable, motion sensor based activity recognition from monocular rgb videos. In *Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers*, pages 699–708, 2019.

[222] Philip Haeusser, Alexander Mordvintsev, and Daniel Cremers. Learning by association–a versatile semi-supervised training method for neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–98, 2017.

[223] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[224] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–30, 2019.

[225] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. Selfhar: Improving human activity recognition through self-training with unlabeled data. *arXiv preprint arXiv:2102.06073*, 2021.

[226] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[227] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[228] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and S Yu Philip. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2018.

[229] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–9. IEEE, 2018.

[230] Xin Du, Katayoun Farrahi, and Mahesan Niranjan. Transfer learning across human activities using a cascade neural network architecture. In *Proceedings of the 23rd international symposium on wearable computers*, pages 35–44, 2019.

[231] Xin Qin, Yiqiang Chen, Jindong Wang, and Chaohui Yu. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–25, 2019.

[232] Francisco Javier Ordóñez Morales and Daniel Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 92–99, 2016.

[233] Valentin Radu and Maximilian Henne. Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–21, 2019.

[234] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1037–1045, 2015.

[235] David C Mohr, Mi Zhang, and Stephen M Schueller. Personal sensing: understanding mental health using ubiquitous sensors and machine learning. *Annual review of clinical psychology*, 13:23–47, 2017.

[236] Sandra Servia-Rodríguez, Kiran K Rachuri, Cecilia Mascolo, Peter J Rentfrow, Neal Lathia, and Gillian M Sandstrom. Mobile sensing at the service of mental well-being: a large-scale longitudinal study. In *Proceedings of the 26th International Conference on World Wide Web*, pages 103–112, 2017.

[237] Heng-Tze Cheng, Feng-Tso Sun, Martin Griss, Paul Davis, Jianguo Li, and Di You. Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. pages 361–374, 06 2013.

[238] Tong Wu, Yiqiang Chen, Yang Gu, Jiwei Wang, Siyu Zhang, and Zhanghu Zhechen. Multi-layer cross loss model for zero-shot human activity recognition. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 210–221, Cham, 2020. Springer International Publishing.

[239] Moe Matsuki, Paula Lago, and Sozo Inoue. Characterizing word embeddings for zero-shot sensor-based human activity recognition. *Sensors*, 19(22):5043, 2019.

[240] Valter Estevam, Helio Pedrini, and David Menotti. Zero-shot action recognition in videos: A survey. *Neurocomputing*, 439:159–175, 2021.

[241] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[242] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[243] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[244] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[245] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[246] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

[247] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5327–5336, 2016.

[248] Zeynep Akata, Honglak Lee, and Bernt Schiele. Zero-shot learning with structured embeddings. *arXiv preprint arXiv:1409.8403*, 2014.

[249] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2927–2936, 2015.

[250] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5327–5336, 2016.

[251] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 69–77, 2016.

[252] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2452–2460, 2015.

[253] Qian Wang and Ke Chen. Zero-shot visual recognition via bidirectional latent embedding. *International Journal of Computer Vision*, 124(3):356–383, 2017.

[254] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[255] Chuanxing Geng, Lue Tao, and Songcan Chen. Guided cnn for generalized zero-shot and open-set recognition using visual and semantic prototypes. *Pattern Recognition*, 102:107263, 2020.

[256] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. *CoRR*, abs/1605.04253, 2016.

[257] Yannick Le Cacheux, Hervé Le Borgne, and Michel Crucianu. From classical to generalized zero-shot learning: a simple adaptation process. *CoRR*, abs/1809.10120, 2018.

[258] Rafael Felix, Michele Sasdelli, Ian D. Reid, and Gustavo Carneiro. Multi-modal ensemble classification for generalized zero shot learning. *CoRR*, abs/1901.04623, 2019.

[259] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, and Xi-Zhao Wang. A review of generalized zero-shot learning methods. *CoRR*, abs/2011.08641, 2020.

[260] J.K. Aggarwal and Lu Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, 2014. Celebrating the life and work of Maria Petrou.

[261] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[262] Heng-Tze Cheng, Martin Griss, Paul Davis, Jianguo Li, and Di You. Towards zero-shot learning for human activity recognition using semantic attribute sequence model. pages 355–358, 09 2013.

[263] David Kahle, Terrance Savitsky, Stephen Schnelle, and Volkan Cevher. Junction tree algorithm. *Stat*, 631, 2008.

[264] Hiroki Ohashi, Mohammad Al-Naser, Sheraz Ahmed, Katsuyuki Nakamura, Takuto Sato, and Andreas Dengel. Attributes' importance for zero-shot pose-classification based on wearable sensors. *Sensors*, 18(8):2485, 2018.

[265] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[266] Sumit Majumder and M Jamal Deen. Smartphone sensors for health monitoring and diagnosis. *Sensors*, 19(9):2164, 2019.

[267] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. *Vid2Doppler: Synthesizing Doppler Radar Data from Videos for Training Privacy-Preserving Activity Recognition*. Association for Computing Machinery, New York, NY, USA, 2021.

[268] A. Saeed, T. Ozcelebi, and J. Lukkien. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)*, 3(2):1–30, 2019.

[269] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 International Symposium on Wearable Computers*, pages 45–49, 2020.

[270] Harish Haresamudram, Irfan Essa, and Thomas Plötz. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–26, 2021.

[271] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 31, 2018.

[272] Shohreh Deldari, Hao Xue, Aaqib Saeed, Daniel V Smith, and Flora D Salim. Cocoa: Cross modality contrastive learning for sensor data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–28, 2022.

[273] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. Handling annotation uncertainty in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*, pages 109–117, 2019.

[274] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[275] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*, 2016.

[276] Karina Sokolova and Charles Perez. You follow fitness influencers on youtube. but do you actually exercise? how parasocial relationships, and watching fitness influencers, relate to intentions to exercise. *Journal of Retailing and Consumer Services*, 58:102276, 2021.

Appendix A

A.1 Collaboration Acknowledgments

The following lists the works within this thesis and provides details on my contributions:

[27] H. Kwon*, C. Tong*, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Plötz. IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 4 Issue 3, Ubicomp'21, 2020.

This work forms the basis of Chapter 3. In this work, H.Kwon and I were equally-contributing authors who coordinated the research efforts in our respective universities (Georgia Institute of Technology and the University of Oxford). Kwon's extensive knowledge and background in computer vision were instrumental to the design of the IMUTube pipeline. My contributions were: motivating the use of visual information for inertial Human Activity Recognition (HAR), proposing and experimenting with different machine learning approaches to improve the realism of virtual IMU data in the IMUTube pipeline (together with Y.Gao), designing evaluation studies to demonstrate the feasibility of virtual IMU data, implementing and reporting deep learning experiments (except the ones on unsupervised transfer learning, which were done by H.Haresamudram), leading the framing and writing of the IMWUT paper. Together with Kwon, I presented this work at Ubicomp 2020, which was nominated for Best Presentation by the judges.

[28] C. Tong and N. D. Lane. Empowering IMU-based Activity Recognition Through Cross-Modal Knowledge Transfer from Unpaired Videos. Work in submission.

This work forms the basis of Chapter 4. My contributions were: proposing and developing the idea of cross-modal knowledge transfer between visual and inertial HAR models, designing the framework and training mechanism to accomplish unpaired transfer from visual to inertial modalities, implementing and reporting all experiments and their empirical findings.

[29] C. Tong*, J. Ge*, and N. D. Lane. Zero-Shot Learning for IMU-Based Activity Recognition Using Video Embeddings. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (IMWUT) Vol. 5 Issue 4, Ubicomp'22, 2021.

This work forms the basis of Chapter 5. In this work, J.Ge and I are equally-contributing authors. My contributions were: proposing the study to assess the use of semantic spaces for zero-shot learning (ZSL) in inertial HAR, proposing the use of visual embeddings as a ZSL semantic space, implementing and reporting experiments and results, leading the framing and writing of the IMWUT paper; I presented this work at Ubicomp 2022. My co-first-author, Ge, contributed to the initial implementation, data collection and experimentation of the study.

A.2 Train-Test Splits for Zero-Shot Learning

This section provides the details of the train-test splits used throughout our Zero-shot Learning work in Chapter 5.

On each of the three denoted datasets, we use k-fold evaluation with a fixed set of seen classes for training and the rest for testing. The test classes for the evaluated PAMAP2, DaLiAc and UTD-MHAD datasets are shown in Table A.1, Table A.2 and Table A.3 respectively.

We have arrived at these train-test splits as follows: We observe that the splits adopted by previous work [101] ensured that at least one activity of each type is present in the training (seen) set and the test (unseen) set, according to the activity types which we have identified in Table A.4. Inspired by this, we construct similar tables for DaLiAc (Table A.5) and UTD-MHAD (Table A.6). We arrive at the current splits by randomly selecting classes of different activity types into the test set.

Fold	Test Classes
1	watching TV, house cleaning, standing, ascending stairs
2	walking, rope jumping, sitting, descending stairs
3	playing soccer, lying, vacuum cleaning, computer work
4	cycling, running, Nordic walking
5	ironing, car driving, folding laundry

Table A.1: Train-test split of the PAMAP2 dataset from Wang *et al.* [101]

Fold	Test Classes
1	sitting, vacuuming, descending stairs
2	lying, sweeping, car driving
3	standing, walking, bicycling on ergometer
4	washing dishes, ascending stairs, rope jumping

Table A.2: Train-test split of the DaLiAc dataset.

Fold	Test Classes
1	swipe left, cross arms, draw triangle, arm curl, jogging in place, pick up then throw
2	swipe right, basketball shoot, bowling, tennis serve, walking in place, squat
3	wave, draw x, boxing, two hand push, sit then stand
4	clap, draw circle clockwise, baseball swing, knock on door, stand then sit
5	throw, draw circle counter clockwise, tennis swing, hand catch, forward lunge

Table A.3: Train-test split of the UTD-MHAD dataset.

A.3 Attribute Space for Zero-Shot Learning

This section documents the attribute-based class prototypes used for Zero-Shot Learning (Chapter 5). The following tables list the attribute vectors that are manually defined for the different activity classes in the PAMAP2 (Table A.7), DaLiAc (Table A.8) and UTD-MHAD (Table A.9) datasets.

Activity Types	Classes
Static activities	lying, sitting, standing
”Walking” activities	walking, Nordic walking, ascending stairs, descending stairs
House chores	vacuum cleaning, ironing, folding laundry, house cleaning
Sports	running, cycling, playing soccer, rope jumping
”Sitting” activities	watching TV, computer work, car driving

Table A.4: Activity types identified in the PAMAP2 dataset.

Activity Types	Classes
Static activities	sitting, lying, standing
House chores	washing dishes, vacuuming, sweeping
”Walking” activities	walking, ascending stairs, descending stairs
Other activities	car driving, bicycling on ergometer, rope jumping

Table A.5: Activity types identified in the DaLiAc dataset.

Activity Types	Classes
One-hand activities	swipe left, swipe right, wave, throw, knock on door, hand catch
Two-hand activities	clap, cross arms, arm curl, two hand push
Drawing activities	draw x, draw circle clockwise, draw circle counter clockwise, draw triangle
Sports	basketball shoot, boxing, baseball swing, tennis swing, tennis serve
Activities involving legs	bowling, pick up then throw, jogging in place, walking in place, sit then stand, stand then sit, forward lunge, squat

Table A.6: Activity types identified in the UTD-MHAD dataset.

Attribute Aspects	Body Movements																													Related Objects and Environments														
Attribute / Activity	motion	static	cyclic motion	intense motion	translation motion	free motion	body vertical	body incline	body horizontal	body forward	body backward	body up	body down	body in place	torso transform	arms motion	arms static	arms bent	arms straight	arms bent-straight transform	hands hold something	legs motion	legs static	legs bent	legs straight	legs bent-straight transform	legs alternate move forward	legs move up and/or down	seat	bike	poles	television	computer	car	stairs	vacuum	iron	clothes	soccer	rope	indoor	outdoor
lying	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
sitting	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
standing	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
walking	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
running	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
cycling	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
Nordic walking	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
watching TV	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	
computer work	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	
car driving	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	
ascending stairs	1	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	
descending stairs	1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	
vacuum cleaning	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
ironing	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0
folding laundry	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0
house cleaning	1	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0
playing soccer	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rope jumping	1	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1

Table A.7: Manual attributes for the PAMAP2 dataset defined by Wang *et al.* [101]

Attribute Aspects	Body Movements																												Related Objects and Environments														
Attribute / Activity	motion	static	cyclic motion	intense motion	translation motion	free motion	body vertical	body incline	body horizontal	body forward	body backward	body up	body down	body in place	torso transform	arms motion	arms static	arms bent	arms straight	arms bent-straight transform	hands hold something	legs motion	legs static	legs bent	legs straight	legs bent-straight transform	legs alternate move forward	legs move up and/or down	seat	bike	poles	television	computer	car	stairs	vacuum	iron	clothes	soccer	rope	ergometer	broom	dishes	indoor	outdoor
Sitting	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Lying	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Standing	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Washing dishes	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
Vacuuming	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0		
Sweeping	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	
Walking	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1		
Ascending stairs	1	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1		
Descending stairs	1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1		
car driving	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Bicycling on ergometer	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Rope jumping	1	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Table A.8: Manual attributes for the DaLiAc dataset.

Attribute Aspects	Body Movements																						Related Objects					
Activity \ Attribute	body always vertical	body inclined	body up	body down	torso transform	arms static	only one arm	two arm	arm(s) has bent	arms straight	arm(s) periodic motion	hands hold something	drawing (incl. swipe)	drawing return to origin	drawing left first	drawing continuous	legs static	legs motion	legs(s) small bent	legs(s) large bent	leg(s) periodic motion	foot relocate	foot no relocate	basketball	baseball	tennis	chair	bowling ball	object away
swipe left	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
swipe right	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
wave	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
clap	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
throw	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
cross arms	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
basketball shoot	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1
draw x	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0
draw circle clockwise	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
draw circle counter clockwise	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
draw triangle	1	0	0	0	0	0	1	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0
bowling	0	1	1	1	1	0	1	1	1	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1
boxing	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
baseball swing	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0	1	0	0	0	1
tennis swing	1	1	0	0	1	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	1
arm curl	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
tennis serve	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	1
two hand push	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
knock on door	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
hand catch	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
pick up then throw	0	1	1	1	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	1
jogging in place	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
walking in place	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
sit then stand	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0
stand then sit	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0
forward lunge	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
squat	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0

Table A.9: Manual attributes for the UTD-MHAD dataset.