

On the Complementarity of Video and Inertial Data for Human Activity Recognition

DISSERTATION

for obtaining the doctoral degree

Doctor of Natural Sciences (Dr. rer. nat.)

presented by
M.Sc. Marius Bock

submitted to the School of Science and Technology
of the University of Siegen
Siegen 2025

Supervisors and first and second appraisers

Prof. Dr. Michael Möller

&

Prof. Dr. Kristof Van Laerhoven

University of Siegen

Third appraiser

Prof. Dr. Cecilia Mascolo

University of Cambridge

Day of the oral examination

28. November 2025

Acknowledgements

Looking back, my time as a PhD student has had a lasting impact on my life. I began this journey with no prior experience in scientific publishing, driven mainly by the joy of experimenting until you find that one good idea. I quickly came to realize that pursuing a PhD presents immense challenges that one needs to overcome – both mentally and physically. In the following, I would like to express my deepest gratitude towards people who have helped me stay grounded and motivated, especially during times when I felt overwhelmed and overworked.

First and foremost, I want to thank my two supervisors, Michael Moeller and Kristof Van Laerhoven. What began as a shot in the dark - a proposed joint project as a last resort to find funding - turned into a one-of-a-kind collaboration and friendship. Your inspiring leadership, built on respect and genuine concern for your PhD students' well-being, shaped my entire experience. Your guidance extended far beyond academic supervision, offering me career advice that always prioritized my personal context and aspirations. For this, I am immensely thankful.

Being part of two wonderful chairs in Siegen, I am grateful to all my colleagues over the years, who made coming to an aging public university building feel a bit like a second home. Special thanks to Sarah, who always had an answer - no matter how obscure the issue - and helped me navigate even the trickiest administrative hurdles. I am also thankful to Ulrich and Christoph, who kept everything running smoothly behind the scenes and were always there when something did not. Lastly, I owe deep thanks to Jovita Lukasik, Alexander Hölzemann, and Zorah Lähner, whose support from the early days (and still today) helped me stay sane in the often bewildering world of scientific publishing.

Thanks to Michael and Kristof, I also had the opportunity to spend five months abroad at the University of Cambridge. To my supervisor there, Cecilia Mascolo, thank you for welcoming me into your team with such warmth and generosity. Your amazing team, and the people who are part of it, made the experience not only intellectually enriching but also genuinely fun. I would also like to thank Daniel Roggen for his continued support into my research, for the many insightful and stimulating conversations, and for giving me the opportunity to visit Google to discuss my work. Lastly, special thanks to Christina Runkel, Daniel Kreuter, and Mathias Ciliberto for helping me settle in during my move to the UK.

Next, I want to thank two people who helped set me on the path to a PhD in the first place. Glenn Neuber, thank you for introducing me to data science during my internship at SAP. Your enthusiasm for machine learning and Kaggle competitions was contagious and helped me discover my own passion for solving prediction problems. Margret Keuper, thank you for your patient and caring support during my master's thesis, and for helping me find my PhD position and introducing me to Michael and Kristof.

Throughout my academic journey, I have been lucky to have amazing friends, both near

and far, who helped me disconnect after long days and reminded me that there is more to life than work. To all the wonderful people I met while living abroad, to my former flatmates, especially Charlotte and Raquel, who made me part of their group of friends when I moved to Siegen, and most of all, to all of my high school friends, who stayed in contact with me throughout the years of always being on the move: I am grateful to have you all in my life, even after all these years of studying.

Being the youngest sibling, my family has always been my anchor - especially during the turbulent moments of my twenties. Ricarda and Torsten: you have always been my role models. Being your little brother always felt like one of the greatest gifts, as I always knew that you were there for advice and gave me the certainty that I could also be as successful and happy in life as you two are when I get older. To my parents, Stefanie and Manfred: words cannot express how grateful I am to have you as my parents. You were the reason why I could always pursue my crazy plans and dreams, supporting them not only financially but also with your unconditional love and encouragement. None of this would have been possible without you.

Finally, I want to thank the person who made Siegen more than just the place where I completed my PhD – my girlfriend, Ann-Kathrin. Meeting you has been by far the greatest gift of this journey. If it weren't for you encouraging me to go on walks, play board games to distract me from my laptop, or reminding me to celebrate when a paper was accepted, I probably would have quit a gazillion times. I know this journey wasn't easy for you either (especially during deadlines), and that you put your own dreams aside for now so that I could pursue mine. Despite all this, you were, and always are, my biggest, most unconditional fan, and for that, I will always be thankful. I love you.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer, nicht angegebener Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Es wurden keine Dienste eines Promotionsvermittlungsinstituts oder einer ähnlichen Organisation in Anspruch genommen.

Zu den vorgeschlagenen Mitgliedern der Promotionskommission bestehen keine verwandtschaftlichen Beziehungen, keine Verwandtschaft ersten Grades, Ehe, Lebenspartnerschaft oder eheähnliche Gemeinschaft.

Marius Bock

Siegen, den 26. Juni 2025

Abstract

With research in fields such as psychiatry having shown strong links between activities and behavior, there has been a growing interest in the development of automatic activity recognition systems using machine learning methods, also known as Human Activity Recognition (HAR). Within the last decade, Deep Learning methods have surpassed classical machine learning models in performance and have become the de facto standard learning-based approach for sensor-based HAR. While Deep Learning has largely automated feature extraction from inertial data, reducing the dependence on expert-crafted features, it has inadvertently introduced new challenges to the activity recognition community. This dissertation, structured in two parts, addresses two of these core challenges associated with applying deep learning to inertial-based HAR by leveraging concepts and methodologies from the domain of computer vision.

The first part of the dissertation focuses on the so-called *labeling bottleneck*, which denotes the considerable manual effort and cost associated with annotating data from wearable inertial sensors. This issue has significantly limited the scale and complexity of publicly available HAR benchmark datasets, thereby negatively affecting methodological progress. In an effort to decrease annotator workload, a weak-annotation pipeline is proposed that only requires labels for representative segments of a synchronously recorded video stream by leveraging the discriminative capabilities of vision foundation models. The second part examines the *sliding window problem*, referring to the temporal modeling limitations caused by HAR approaches relying on fixed-length window classification. Showcasing a reformulated view to inertial-based HAR, this dissertation introduces vision-based Temporal Action Localization (TAL) into the inertial domain. Benchmark experiments demonstrate that both existing TAL models from the video domain and a newly proposed TAL-inspired architecture for inertial data significantly outperform classical inertial HAR models. By leveraging inter-segment temporal context, both approaches also exhibit reduced sensitivity to hyperparameters selected during segmentation.

The demonstrated use cases show how recent advancements in video-based activity recognition can help overcome limitations inherent to inertial sensing. While each approach exhibits certain constraints, these works offer novel perspectives on long-standing issues and introduce methodologies that, if adopted, could inspire further research and innovation within the inertial HAR community.

Zusammenfassung

Forschung in Disziplinen wie der Psychiatrie hat enge Zusammenhänge zwischen menschlichen Aktivitäten und Verhaltensweisen aufgezeigt, was zu einem wachsenden Interesse an der Entwicklung automatisierter Systeme zur Aktivitätserkennung auf Basis maschineller Lernverfahren geführt hat – auch bekannt als Human Activity Recognition (HAR). Im Verlauf der letzten Dekade haben Deep-Learning-Methoden klassische maschinelle Lernverfahren in ihrer Leistungsfähigkeit übertroffen und sich als De-facto-Standard für sensorbasierte HAR-Systeme etabliert. Während Deep Learning die Merkmalsextraktion aus inertialen Sensordaten weitgehend automatisiert und damit die Abhängigkeit von expertengestützten, manuell entwickelten Merkmalen reduziert hat, bringt dieser Paradigmenwechsel zugleich neue Herausforderungen für die HAR-Community mit sich. Diese Dissertation, gegliedert in zwei Hauptteile, widmet sich zwei dieser grundlegenden Herausforderungen, die mit dem Einsatz von Deep Learning in der inertialbasierten HAR verbunden sind, und greift dabei auf Konzepte und Methoden aus dem Bereich der Computer Vision zurück.

Der erste Teil der Arbeit befasst sich mit dem sogenannten *Labeling Bottleneck*, das den erheblichen manuellen Aufwand und die damit verbundenen Kosten bei der Annotation von Daten tragbarer inertialer Sensoren bezeichnet. Diese Problematik hat den Umfang und die Komplexität öffentlich verfügbarer HAR-Benchmark-Datensätze erheblich eingeschränkt und dadurch den methodischen Fortschritt negativ beeinflusst. Zur Reduktion des Aufwands für Annotator*innen wird eine Weak-Annotation-Pipeline vorgeschlagen, die lediglich die Annotation repräsentativer Segmente eines synchron aufgezeichneten Videostreams erfordert. Hierbei werden die diskriminativen Fähigkeiten moderner Vision Foundation Models genutzt. Im zweiten Teil wird das *Sliding Window Problem* behandelt, das sich auf die begrenzten Möglichkeiten zur zeitlichen Modellierung in HAR-Ansätzen bezieht, die auf der Klassifikation von Sequenzen fester Länge basieren. Im Sinne einer Neuperspektivierung inertialbasierter HAR werden Konzepte der videobasierten Temporal Action Localization (TAL) auf den inertialen Bereich übertragen. Anhand von Benchmark-Experimenten wird gezeigt, dass sowohl bestehende TAL-Modelle aus dem Videobereich als auch eine eigens entwickelte, TAL-inspirierte Architektur für inertielle Daten klassische inertialbasierte Modelle signifikant übertreffen. Durch die Nutzung intersegmentaler zeitlicher Kontextinformationen zeigen beide Ansätze eine verringerte Sensitivität gegenüber den bei der Segmentierung gewählten Hyperparametern.

Die dargestellten Anwendungsfälle zeigen, wie aktuelle Fortschritte in der videobasierten Aktivitätserkennung dazu beitragen können, inhärente Einschränkungen der Inertialsensorik zu überwinden. Auch wenn jede Herangehensweise bestimmte Grenzen aufweist, bieten diese Arbeiten neue Perspektiven auf langjährige Probleme und führen Methoden ein, die, bei entsprechender Anwendung, weitere Forschung und Innovation innerhalb der inertialen HAR-Community anregen könnten.

Contents

1. Introduction	1
1.1. Motivation & Contributions	1
1.2. Outline	7
1.3. List of Publications	8
2. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition	10
2.1. Introduction & Motivation	13
2.2. Related Work	14
2.3. Methodology	18
2.4. Benchmarks and Baseline Results	22
2.5. Ablation Experiments	29
2.6. Limitations	32
2.7. Discussion & Conclusion	34
3. Weak-Annotation of HAR Datasets using Vision Foundation Models	36
3.1. Introduction	38
3.2. Related Work	39
3.3. Methodology	41
3.4. Results	44
3.5. Discussion & Conclusion	48
4. Improving Deep Learning for HAR with shallow LSTMs	50
4.1. Introduction	52
4.2. Related Work	53
4.3. Methodology	54
4.4. Conclusion and Future Directions	58
5. Temporal Action Localization for Inertial-based Human Activity Recognition	60
5.1. Introduction	62
5.2. Related Work	64
5.3. Temporal Action Localization for Inertial-based HAR	65
5.4. Methodology	70
5.5. Results	73
5.6. Discussion & Conclusion	80

6. DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activitiy Recognition	84
6.1. Introduction	86
6.2. Related Work	87
6.3. Methodology	88
6.4. Results	91
6.5. Limitations	95
6.6. Discussion & Conclusions	97
7. Conclusion	98
Bibliography	101
Appendix	127

List of Acronyms

HAR	Human Activity Recognition
IMU	Inertial Measurement Unit
ARC	Activity Recognition Chain
DL-ARC	Deep Learning Activity Recognition Chain
TAL	Temporal Action Localization
LOSO	Leave-One-Subject-Out
mAP	mean Average Precision
tIoU	temporal Intersection over Union
GMM	Gaussian Mixture Models
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
CNN	Convolutional Neural Network

1. Introduction

1.1. Motivation & Contributions

With research in fields such as psychiatry having shown strong links between activities and behavior [173], there has been a growing interest in the development of automatic activity recognition systems using machine learning methods, commonly referred to as Human Activity Recognition (HAR) [26]. Particularly Inertial Measurement Units (IMUs), consisting of an accelerometer, gyroscope and magnetometer, have established themselves as a popular choice for implementing HAR systems, as they offer a robust form of data collection to continuously observe motion and gestures at particular body locations with minimal impact on the user [183, 128, 137]. In 2014, Bulling et al. [26] released a well-received tutorial on HAR from wearable sensor data, introducing the Activity Recognition Chain (ARC) – a step-by-step framework for designing and evaluating inertial-based activity recognition systems. As early HAR systems primarily relied on classical machine learning methods, the original ARC did not account for Deep Learning approaches. However, the surge in popularity of Deep Learning in fields such as Computer Vision and Natural Language Processing [94, 37, 163] has led researchers to investigate the effectiveness of these methods to HAR prediction problems [126]. Since the release of the original ARC, Deep Learning methods have surpassed classical machine learning models in performance and popularity and have become the de facto standard learning-based approach for sensor-based HAR – largely because of their automated feature extraction capabilities which previously required expert knowledge. To reflect this methodological shift, an updated perspective on the ARC was developed as part of this dissertation, resulting in the Deep Learning Activity Recognition Chain (DL-ARC), presented in Figure 1.1. The DL-ARC is structured into three main phases:

1. *Data collection & annotation*: This phase involves recruiting participants and defining a sensor layout to record (body-worn) sensor data. Participants are asked to perform a predefined set of activities C while wearing a selection of inertial sensors. To ensure consistency, data is often collected in controlled lab environments using standardized protocols. As inertial signals are not human-interpretable, video recordings are commonly used to enable accurate offline activity annotation.
2. *Preprocessing & segmentation*: After data recording and annotation, additional prepro-

cessing steps are typically needed before inputting the data into Deep Learning models. These may include handling different sensor sampling rates, imputing missing values, removing outliers, and normalizing the data [52]. The result is a synchronized, multi-sensor timeline $X \in \mathbb{R}^{N \times S}$ with corresponding activity labels $Y \in \mathbb{R}^N$, where N is the number of records and S the number of sensor axes. A sliding window w of size $W \times S$ is then applied to X with stride s , producing a discretized version $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, where each window $\mathbf{x}_t \in \mathbb{R}^{W \times S}$ corresponds to a timestamp $t = \{0, 1, \dots, T\}$ and is assigned a label $a_t \in \{1, \dots, C\}$, often using majority voting over the corresponding segment in Y .

3. *Classification & evaluation*: Conceptually, inertial activity recognition is formulated as a multi-class classification task, whose goal is to approximate an underlying function $\mathcal{G}(x) = y$, mapping any window of inertial data $\mathbf{x}$ to its corresponding activity label a in C , using a parametrized function $\mathcal{N}(x, \theta)$. Training an inertial activity recognition system involves determining a set of suitable weights θ such that

$$\mathcal{N}(x_t, \theta) \approx a_t \quad (1.1)$$

holds for all of our collected pairs of sliding windows and activity labels (x_t, a_t) . Specifically, we try to ensure that this property also holds for any new window-activity pair (x_t, a_t) , previously unseen during training of the system, such that ultimately

$$\mathcal{N}(x_t, \theta) \approx \mathcal{G}(x). \quad (1.2)$$

To optimize our parametrized function inertial models typically output class-wise probability vectors $p(a_t)$, which are used to calculate a loss such as cross-entropy loss for backpropagation. Finally, predicted labels can be obtained via $\arg \max p(a_t)$ and compared against the ground truth activity labels to calculate evaluation metrics such as accuracy or F1-score.

Though research within the last decade has been shown to advance the field of Deep Learning for HAR, there are still persisting issues along the DL-ARC. This dissertation aims to address two of these issues as part of its core contributions, namely the *labeling bottleneck* and the *sliding window problem*, by combining recent advancements within Computer Vision with inertial activity recognition. The following will briefly outline both problems and provide an overview of how the published works presented in this dissertation contribute to solving them within the inertial-based HAR community, along with highlighting additional impacts achieved during the course of this PhD.

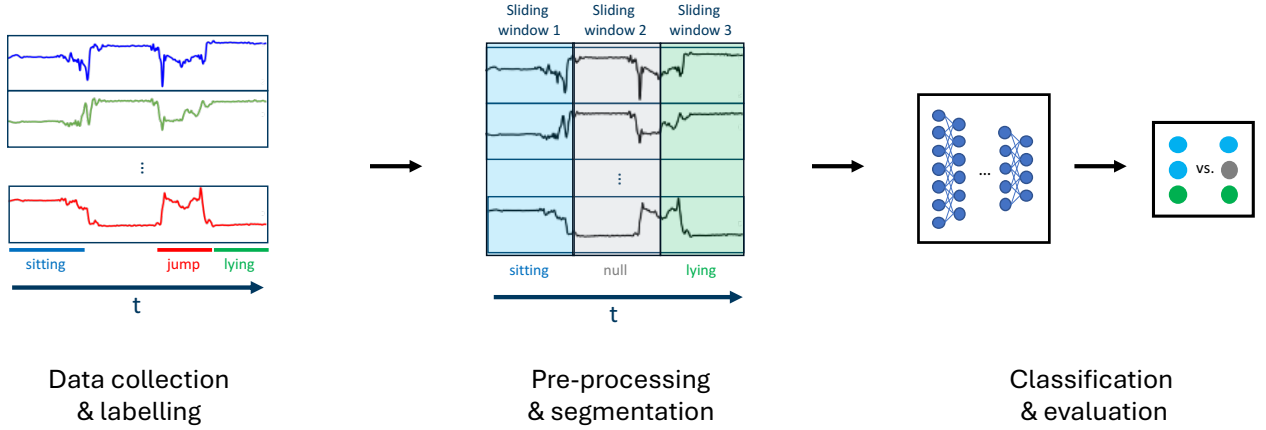


Figure 1.1.: Deep Learning Activity Recognition Chain (DL-ARC) developed as part this dissertation project and introduced during a tutorial at an international conference. This updated pipeline extends the original ARC [26] to incorporate the shift from classical machine learning to Deep Learning techniques.

The Labeling Bottleneck. The lack of interpretability of wearable sensor data in hindsight as well as an increased risk of errors introduced by self-reported annotations of participants, has led researchers to rely on video data as means of retrospectively annotating their HAR experiments (see e.g. [142]). Though initial investigations conducted as part of this dissertation have shown that wearable data annotation can be outsourced to novice annotators by means of annotation tools that show video and inertial data streams simultaneously, the lack of such tools as well as remaining high workloads usually require researchers annotating benchmark datasets themselves [76]. This *labeling bottleneck* of acquiring sufficient amounts of annotated data, has caused benchmark datasets used within the field of HAR to lack in richness and size compared to that of related fields. Though Deep Learning has allowed HAR researchers to automatically extract discriminative features from raw input sensor data, these methods have been shown to require large amounts of input data in order to properly converge [94, 8, 70, 74]. Though learning paradigms such as self-supervised learning [145, 166, 198, 67], data augmentation [211] or synthetic data generation [96, 97, 138] have emerged as a way to circumvent the need for annotated data, these methods have shown limitations in replacing the inherent diversity and nuances of real-world data [139, 105] as well as do not replace the need for proper assessment on downstream tasks which might be a critical requirement for task-specific activity recognition systems deployed in-the-wild.

With a majority of currently publicly-available inertial datasets reported to rely primarily on video recordings for offline labeling (e.g. [201, 142, 149]), Chapter 3 proposes a method that can be used to automatically analyze such experimental videos as means to significantly lower the required annotation effort. The proposed pipeline combines a clustering-based annotation approach with pretrained vision foundation models such as CLIP, which have shown

to help the vision community advance in finding robust, generalizable feature representations. Instead of being tasked to manually annotate each timestamp within an HAR experiments, the pipeline asks annotators to only provide activity labels for representative clips within a video. These so called "centroid clips" are determined by clustering latent representations of the clips extracted using vision foundation models. The user-provided label information is then propagated to all other clips within a cluster, resulting in a weakly-annotated version of the dataset with minimal annotation effort. As part of this core contribution, Chapter 2 introduces a novel benchmark dataset, called WEAR, which provides both inertial and egocentric video data. WEAR uses a sensor layout in-line with best-practices of both the vision and inertial community, setting it apart large corpora such as Ego4D [59] and is collected under diverse, outdoor recording conditions. Combining the sensor layout of WEAR, which does not restrict participants to the field-of-view of an experimental camera or an artificial lab-environment, with the annotation pipeline, this dissertation draws a data collection scheme in which participants could collect in-the-wild data and use their own synchronized videos for reliable self-annotation, reducing participant burden and possibly improving long-term study retention.

The Sliding Window Problem. As already discussed above, due to the temporal nature of activities, early works of HAR research have relied on pre-segmenting timelines, training activity recognition systems to classify a collection of records, called sliding window, rather than individual ones. Being applied to classify windows in $\mathcal{X}$ individually, models such as the DeepConvLSTM [126] are restricted to activity patterns and temporal context present within only the window itself. Known as the *sliding window problem*, window size w and stride s thus need to be set carefully, as setting them incorrectly could result in significant performance drops in recognition performance. If dealing with both long and short lasting activities, setting w too small might end up windows not to fully capturing temporal sequences of an activity, while setting it too large might cause sliding windows containing mixed activity types, yet being associated only with one activity label.

Chapters 4, 5 and 6 aim to address the sliding window problem by investigating temporal modeling capabilities of inertial architectures and proposing reformulations to the classification approach described above. First Chapter 4 shows that parts of inertial networks responsible for temporal learning, specifically LSTMs, can be simplified by using shallower variants. Results obtained using the classic DeepConvLSTM [126] show that while this reduces the complexity of the model, prediction performance significantly improves across multiple benchmark datasets. Contradicting the common belief that one needs at least two recurrent layers when dealing with sequential data [89], this work questions the current trends within the HAR community towards more complex temporal modeling techniques.

Second, while benchmarking the WEAR dataset (see Chapter 2) it was noted that vision-based Temporal Action Localization (TAL) models can be applied to inertial data. By means of simple vectorization, sliding windows of inertial data can serve as input to these models. Chapter 5 thus presents a follow-up work which shows that TAL model outperform classic inertial architectures on a multitude of datasets by significant margin. Similar to inertial activity recognition, vision-based TAL also aims to recognize activities within a recorded timeline. Nevertheless, though inherently sharing a mutual goal, the TAL community does not rely on the classification of sliding windows to localize activities, but aims to predict activity segments within a timeline by following a sequence labeling problem formulation for action localization (see Figure 1.2). That is, in order to predict activity segments $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, where N varies across participants and $\mathbf{y}_i = (s_i, e_i, a_i)$ is defined by its starting time $s_i \in [0, T]$ (onset), end time $e_i \in [0, T]$ (offset) and its associated activity label $a_i \in \{1, \dots, C\}$, TAL models aims to learn to label a sequence of one-dimensional feature embeddings $\mathcal{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_T\}$ extracted from the original sliding windows $\mathcal{X}$ as

$$\mathcal{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_T\} \rightarrow \hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_T\}, \quad (1.3)$$

where $\hat{\mathbf{y}}_t = (p(a_t), d_t^s, d_t^e)$ at timestamp t is defined by a probability vector $p(a_t)$ indicating the class-wise probability of the timestamp being classified as one of the activity categories C and $d_t^s > 0$ and $d_t^e > 0$ being the distance between the current timestamp t and the current segment's onset and offset. Note that d_t^s and d_t^e are not defined if the timestamp is to be classified as background. Finally, the sequence labeling formulation can then be easily decoded to activity segments with

$$a_t = \arg \max p(a_t), \quad s_t = t - d_t^s, \quad \text{and} \quad e_t = t + d_t^e, \quad (1.4)$$

resulting in a collection of segment proposals and associated confidence scores that can be used in combination with e.g. Non-Maximum Suppression to reconstruct the activity timeline. By processing entire timelines rather than isolated windows as well as making each window predict its relative position in the segment its residing in, TAL models are capable of learning both local as well as global temporal patterns that extend across the temporal horizon of a single sliding window. Chapter 5 thus offers a compelling solution to the long-standing sliding window problem, with the TAL community offering many unexplored design choices and training concepts for a multitude of application scenarios, which could be of interest to the inertial-based community.

Lastly, while the suggested architecture in Chapter 4 reduces complexity of the DeepConv-LSTM, the work also introduces, yet does not discuss, a novel form of inter-window temporal learning. By applying recurrent layers within the model across the sequence of windows in

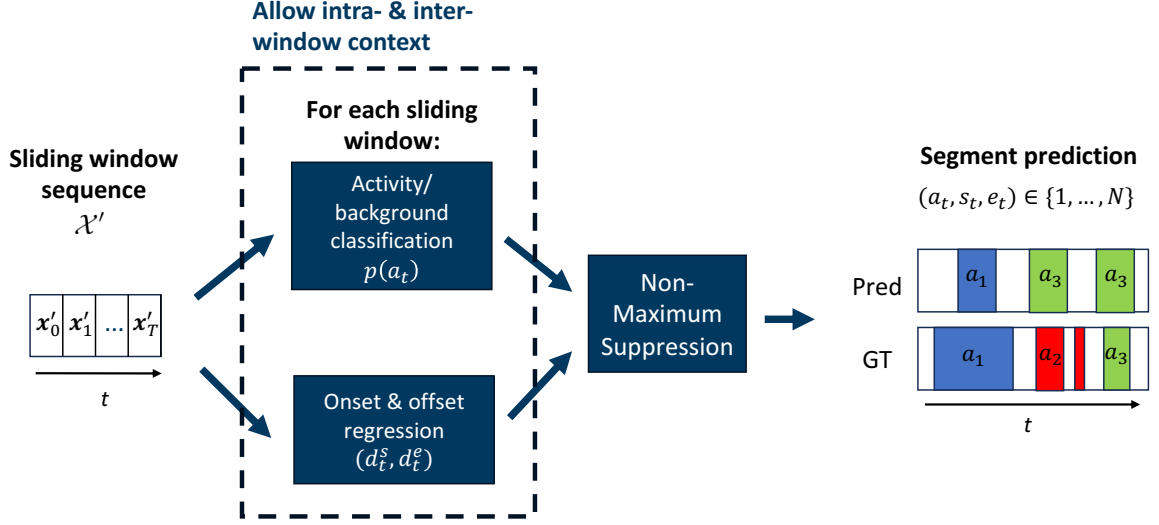


Figure 1.2.: Overview of the applied sequence labeling formulation applied by single-stage TAL models investigated in Chapter 5. Using a sequence of feature embeddings $\mathcal{X}'$ extracted from sliding windows $\mathcal{X}$ as input, for each sliding window, models aim to predict a probability distribution vector $p(a_t)$ as well as the windows onset d_t^s and offset d_t^e . After decoding the sequence, segment proposals and associated confidence scores of all windows are collected and used to reconstruct the activity timeline using e.g. Non-Maximum Suppression.

a batch rather than across the convolutional features within a single window, investigated models are capable of learning inter-window rather than intra-window temporal dependencies. Though deviating from conventional batch semantics, the model still achieves strong recognition performance, suggesting the potential value of inter-window learning. Combining insights from both Chapters 4 and 5, Chapter 6 proposes a novel architecture for inertial-based HAR, called DeepConvContext. The architecture introduces a novel multi-scale approach to inertial activity recognition which extends the classic DeepConvLSTM to learn both inter- and intra-window context. Benchmark experiments show that DeepConvContext outperforms both the original and shallower variant of the DeepConvLSTM Chapter 4 proposes, by a significant margin. Ablation experiments further highlight the superiority of LSTMs as compared to attention to learn temporal patterns in HAR. Though both TAL models as well as DeepConvContext still show certain limitations, especially regarding online prediction capabilities and edge-device suitability, their design choices can be easily translated to other existing works. These design choices are thus likely to inspire further improvements by other researchers, ultimately helping to mitigate the impact that segmentation of inertial data has on model prediction capabilities.

Other contributions. Besides contributing to the aforementioned two sub-problems of activity recognition, other efforts have been made as part of this PhD to contribute to the

HAR community. First, to engage in open discussions and summarize lessons learned as a community, the release of the DL-ARC was done as a full-day tutorial at the ACM International Joint Conference on Pervasive and Ubiquitous Computing and the ACM International Symposium on Wearable Computing (UbiComp/ ISWC). Having been well-received, the tutorial was succeeded by other researchers, continuing and refining the tutorial initial efforts with an extended focus on specific issues along the DL-ARC [68, 69]. Secondly, in an effort to promote reproducibility code of all publications mentioned in this dissertation is open-source, documented and maintained on GitHub. This allows other researchers to easily get in contact if parts of the implementation and released datasets are not working as expected or can be improved in documentation. Lastly, my PhD included volunteering in various organizing committees, namely as Student Volunteer Co-Chair at UbiComp/ ISWC in 2023, organizing committee of the International Workshop on Human Activity Sensing Corpus and Applications (HASCA) hosted at UbiComp/ ISWC as well as technical program committee of ISWC both from 2024 onwards. In addition to being part of the organizing committee of HASCA, I am also the main organizer of a yearly hosted dataset challenge surrounding the WEAR dataset introduced in Chapter 2, whose results are presented and published as part of the HASCA workshop.¹

1.2. Outline

This cumulative dissertation represents central parts of the corpus of work published as PhD student at the University of Siegen, alongside [15, 129, 16, 77, 75, 76, 18]. Central part of this dissertation are four recently published, internationally peer-reviewed publications [17, 20, 22, 23] as well as a preprint [21], which at time of submitting this dissertation is still under review. This dissertation presents each of these central works as an individual chapter in the original text of their respective final versions, with updated formatting and minor corrections. Every chapter is preceded by a discussion section, containing a brief overview over each project, conceptualizing the research done in the context of this thesis and delineating the contributions of the main author and many helpful collaborators who graciously contributed to this research. In this manner the dissertation follows the statutory provisions for a cumulative dissertation laid out in the *Promotionsordnung der Naturwissenschaftlich-Technischen Fakultät* of the University of Siegen. The appendix of this dissertation contains additional unpublished material that may be of interest to specialized readers.

¹WEAR dataset challenge website: mariusbock.github.io/wear/challenge.html

1.3. List of Publications

The following contributions have been made as first-author or equally contributing first-author (denoted by *) as part of this dissertation. All publications are peer-reviewed:

- **Marius Bock**, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. *Improving Deep Learning for HAR with Shallow LSTMs*. In Proceedings of the ACM International Symposium on Wearable Computers. DOI: 10.1145/3460421.3480419 [best paper award]
- **Marius Bock**, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. *Tutorial on Deep Learning for Human Activity Recognition*. Full-day tutorial held at ACM International Joint Conference on Pervasive and Ubiquitous Computing and the ACM International Symposium on Wearable Computing. DOI: 10.48550/arXiv.2110.06663
- Lloyd Pellatt*, **Marius Bock***, Daniel Roggen, and Kristof Van Laerhoven. 2022. *A Public Repository to Improve Replicability and Collaboration in Deep Learning for HAR*. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events. DOI: 10.1109/PerComWorkshops53856.2022.9767436
- **Marius Bock**, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2022. *Investigating (re)current state-of-the-art in human activity recognition datasets*. Frontiers in Computer Science 4, 924954. DOI: 10.3389/fcomp.2022.924954
- Alexander Hoelzemann*, **Marius Bock***, Ericka Andrea Valladares Bastías, Salma El Ouazzani Touhami, Kenza Nassiri, and Kristof Van Laerhoven. 2023. *A Data-Driven Study on the Hawthorne Effect in Sensor-Based Human Activity Recognition*. In Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and the ACM International Symposium on Wearable Computing. DOI: 10.1145/3594739.3610743
- Alexander Hoelzemann*, **Marius Bock***, and Kristof Van Laerhoven. 2024. *Evaluation of Video-Assisted Annotation of Human IMU Data Across Expertise, Datasets, and Tools*. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events. DOI: 10.1109/PerComWorkshops59983.2024.10503292
- **Marius Bock**, Kristof Van Laerhoven, and Michael Moeller. 2024. *Weak-Annotation of HAR Datasets using Vision Foundation Models*. Proceedings of the ACM International Symposium on Wearable Computers. DOI: 10.1145/3675095.3676613
- **Marius Bock**, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. 2024. *WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 8, 4, Article 175. DOI: 10.1145/3699776
- **Marius Bock**, Michael Moeller, and Kristof Van Laerhoven. 2024. *Temporal Action Localization for Inertial-based Human Activity Recognition*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 8, 4, Article 174. DOI: 10.1145/3699770

The following contributions have been made as first-author or shared first-author (denoted by *), but as of handing in this dissertation are still under review within a peer-reviewing process and thus are only provided as preprints:

- **Marius Bock***, Maximilian Hopp*, Kristof Van Laerhoven, and Michael Moeller. 2025. *Label Leakage in Federated Inertial-based Human Activity Recognition*. CoRR abs/2505.20924. DOI:10.48550/arXiv.2505.20924
- **Marius Bock**, Michael Moeller, and Kristof Van Laerhoven. 2025. *DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activity Recognition*. CoRR abs/2505.20894. DOI: 10.48550/arXiv.2505.20894

Additionally the following contributions have been made as co-author:

- Alexander Hoelzemann, Julia L. Romero, **Marius Bock**, Kristof Van Laerhoven, and Qin Lv. *Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors*. MDPI Sensors, 23, 5879. DOI: 10.3390/s23135879
- Sixuan Wu, Alexander Hoelzemann, **Marius Bock**, Kristof Van Laerhoven, Thomas Ploetz, and Alexander T. Adams. *Basketball Shooting Performance Analysis Using Multi-Modal Wearable and Mobile Sensing in Semi-Naturalistic Settings*. In Proceedings of the IEEE 20th International Conference on Body Sensor Networks. DOI: 10.1109/BSN63547.2024.10780653

2. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition

Title: WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition

Authors: Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, Michael Moeller

Venue: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Volume 8, Issue 4

Publication date: 21 November 2024

DOI: 10.1145/3699776

Code: mariusbock.github.io/wear

This work introduces the WEAR dataset, an outdoor sports dataset which provides both inertial and egocentric video data of 22 participants performing 18 workout activities. Unlike previous datasets does WEAR place inertial as well as camera sensor in-line with real-world application scenarios, making it a suitable dataset to assess methods on how to combine strengths of both modalities. While benchmarking popular vision-based TAL models it showed that these models can process raw inertial data and effectively fuse multimodal information. This led to a follow-up study exploring TAL models across various inertial benchmark datasets (see Chapter 5). WEAR has since become a well-established benchmark in both vision- and inertial-based activity recognition communities. Its flexible data collection protocol allows for a yearly dataset challenge at the International Workshop on Human Activity Sensing Corpus and Applications (HASCA), to address current key HAR topics like robustness and multi-sensor fusion.

Contextualization

The first two chapters of this dissertation present two works jointly addressing the *labeling bottleneck* witnessed in the HAR community (see Chapter 1). As part of a DFG project, which partly funded this PhD (grant number 506589320), initial efforts aimed to overcome the lack of interpretability of inertial sensors through the automatic analysis of time-synchronized video data (see Chapter 3). Although previous studies such that of Spriggs et al. [155] have shown complementary strengths between video and inertial data, an initial

investigation upon datasets which provide both modalities showed three major limitations:

1. Though datasets such as Epic-Kitchens [38], Ego4D [59] and Ego-Exo4D [60] provide a combined 1,000 hours of inertial and video data, inertial sensors used in these large corpora capture only head movements from internal camera sensors, offering limited complementary information for learning-based fusion approaches.
2. With the intended audience being biased towards the vision-based community, many datasets [39, 38, 59, 60, 42] are favored towards vision-centric prediction scenarios, with activities happening mostly within the field-of-view of participants and/or being defined by interaction with objects.
3. Lastly, many datasets are recorded in a single, non-changing indoor environment [46], typically being artificially created within laboratory-like environments [39, 42]. While this limits the room for errors and allows for complex sensor-setups, it inherently limits each dataset’s amount of variety captured in the visual data, as lighting conditions and surroundings remain the same throughout all participants.

Considering these limitations, it was quickly established that in order to properly assess the aforementioned use case, a novel dataset would be required which addresses the aforementioned gaps. The following work thus introduces WEAR, an outdoor sports dataset, which provides inertial and egocentric video data of 22 participants each performing a set of 18 workout activities while being equipped with four inertial sensors (one on each limb) and a head-mounted camera to record the participant’s egocentric view. In an effort to provide a challenging prediction scenario with a sensor layout which is in-line with best practices of both the inertial and video activity recognition community, the dataset is marked by purposely introduced activity variations and an overall small information overlap between the inertial and vision data. Furthermore, WEAR was collected across multiple outdoor locations over several months, offering diverse visual contexts with varying weather and environmental conditions.

The work includes a comprehensive benchmark evaluation of popular inertial and vision-based activity recognition methods. For inertial data, we evaluated Attend-and-Discriminate [1] and the Shallow DeepConvLSTM, the latter being developed as part of this dissertation and discussed in Chapter 4. For vision, we used the state-of-the-art single-stage TAL models ActionFormer [204] and TriDet [151]. While TAL models traditionally detect temporal segments directly from untrimmed videos, inertial models classify pre-segmented windows (see Chapter 1). Remarkably, TAL models demonstrated versatility by effectively processing raw inertial data and fusing multimodal inputs in a plain fashion. Using a simple preprocessing

step of vectorizing the two-dimensional, windowed inertial data allowed TAL models to be applied on the previously unexplored modality and led to a follow-up publication [22] which will be discussed in Chapter 5. Besides facilitating for subsequent research described in Chapter 3, WEAR provides both communities (inertial- and vision-based HAR) a common, challenging benchmark dataset to assess the applicability of combined activity recognition approaches.

The need to collect a novel dataset was suggested by Marius Bock upon initial investigation of benchmark datasets and identified shortcomings of existing corpora. Hilde Kuehne provided expertise on vision-based activity recognition and suggested models to use for benchmarking the vision modality. While presenting and discussing early benchmarking results on the WEAR dataset, Michael Moeller identified the novelty in pursuing TAL models for inertial-based HAR, ultimately resulting in the publication presented in Chapter 5. Marius Bock conducted the data collection, literature review and benchmark analysis. Kristof Van Laerhoven contributed expertise in inertial activity recognition, helping Marius Bock in refining the final sensor setup and recording scenario. Hilde Kuehne, Michael Moeller and Kristof Van Laerhoven offered feedback throughout the writing process to enhance the clarity and readability of the final article. This work was further refined through the feedback of anonymous *IMWUT* reviewers, who encouraged the inclusion of a held-out test set, employing Leave-One-Subject-Out cross-validation, and highlighting possible applications and implications of the employed TAL models. Their suggestions helped further improve the experiments and contextualization of the study.

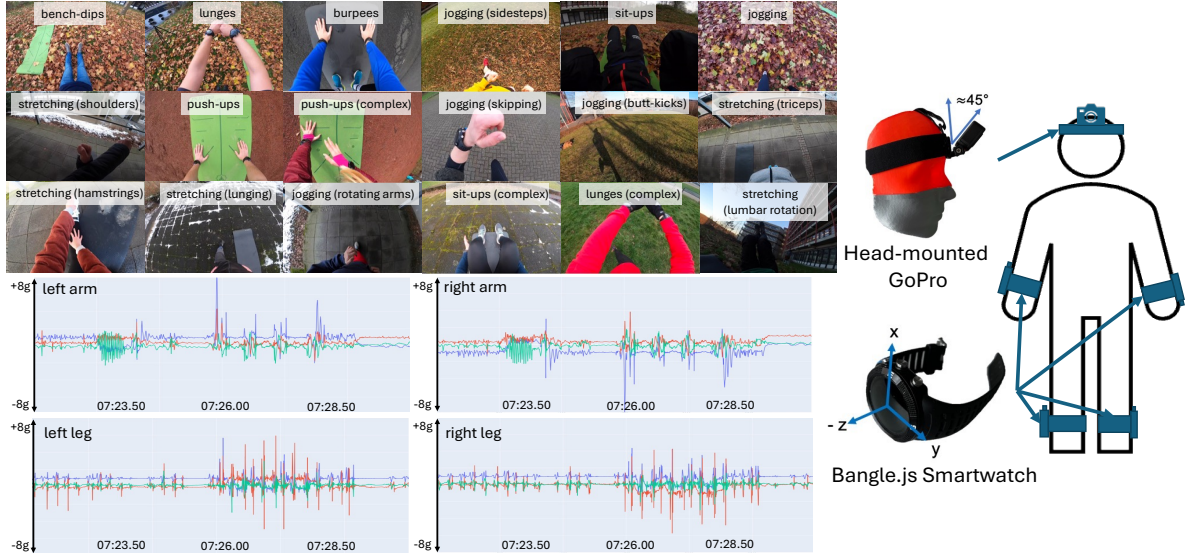


Figure 2.1.: Overview of the sensor setup and example data of the two types of wearable sensors employed in our WEAR dataset. 22 participants each performed 18 activities at 11 different outdoor locations while being equipped with four open-source smartwatches (one per limb) and a head-mounted camera.

Abstract

Research has shown the complementarity of camera- and inertial-based data for modelling human activities, yet datasets with both egocentric video and inertial-based sensor data remain scarce. In this paper, we introduce WEAR, an outdoor sports dataset for both vision- and inertial-based HAR. Data from 22 participants performing a total of 18 different workout activities was collected with synchronized inertial (acceleration) and camera (egocentric video) data recorded at 11 different outside locations. WEAR provides a challenging prediction scenario in changing outdoor environments using a sensor placement, in line with recent trends in real-world applications. Benchmark results show that through our sensor placement, each modality interestingly offers complementary strengths and weaknesses in their prediction performance. Further, in light of the recent success of single-stage TAL models, we demonstrate their versatility of not only being trained using visual data, but also using raw inertial data and being capable to fuse both modalities by means of simple concatenation. The dataset and code to reproduce experiments is publicly available via: mariusbock.github.io/wear/.

2.1. Introduction & Motivation

The physical activities that we perform in our daily lives have been identified as valuable information for a number of research fields and applications [10, 128, 183]. Research efforts

have shown that physical activities can be detected using either wearable inertial sensors or camera-based approaches. The inertial sensors can continuously observe motion and gestures at particular body locations, whereas camera-based systems can typically observe the user’s entire body, but can be hindered by (self-)occlusions. Inertial data manifests itself as multidimensional time series, but is unlike image data almost impossible to be interpreted by a human annotator after recording. Even though research has shown that both modalities are complementary to each other [155], available benchmark datasets that provide both camera and inertial-based sensor data remain scarce. We therefore introduce WEAR, an outdoor sports HAR dataset featuring workout activities performed by 22 participants while wearing inertial sensors on both wrists and ankles as well as a head-mounted camera capturing egocentric vision - see Figure 2.1. Unlike previous egocentric datasets [59, 38, 60], WEAR provides an inertial sensor placement which is in line with recent trends in real-world application scenarios, providing complementary information to the captured video stream. Further, amongst egocentric datasets which utilize limb-placed IMU sensors, WEAR is the first dataset being collected at different outdoor recording locations, with each location introducing different visual and surface conditions, yet not providing cues about the activity being performed. Our contributions in this paper are three-fold:

1. We introduce a new inertial- and vision-based HAR dataset, called WEAR, consisting of data from 22 participants performing 18 different outdoor sports activities.
2. Results of our performed benchmark analysis reveal that the application-driven sensor placement along with the challenging outdoor recording scenario makes WEAR an a suitable dataset to assess methods on how to combine strengths of both inertial- and vision-based approaches.
3. We demonstrate that state-of-the-art TAL models from computer vision are particularly suited to not only process raw inertial data, but even successfully fuse multimodal information significantly outperforming the best single-modality approach as well as beating the best possible (oracle) late fusion approach in terms of mean Average Precision (mAP).

2.2. Related Work

2.2.1. Inertial- and Video-based HAR Datasets

In Table 2.1 we show a curated list of datasets which provide both egocentric vision- (e.g. RGB, depth) and IMU-based (e.g. accelerometer, gyroscope, magnetometer) modalities in

the context of HAR. We compare datasets regarding their recency, size, number of participants, number and type of activities performed, recording environment, limb placement position of IMU sensors and whether the dataset is provided on a clip-basis or a continuous stream. The three largest multimodal corpora currently available, the Ego4D [59], EPIC-Kitchens [38] and Ego-Exo4D [60] datasets only provide inertial data from the built-in IMU of the head-worn camera. Given the captured inertial data thus only decodes the head movement of participants, this position of an inertial sensor is likely to provide only limited complementary information to the egocentric vision modality. As evident by the rise in popularity of commercial head-mounted cameras and wrist-worn smartwatches for tracking sports, we thus decided to position the camera and IMU sensors used during collection of the WEAR dataset in line with the recent trends in real-world application scenarios. With the head and limbs being positions which do not limit participants in their freedom of movement, we deem said positions to further be most suited in capturing how participants interact with their environment and/ or objects. Further note that with the intended audience of the datasets residing mostly in the vision-based community, the CMU-MMAC [39], Ego4D, Ego-Exo4D and EPIC-Kitchens datasets are focusing on providing activity annotations following a vision-centric definition, e.g. *pick up object X*, or are close to a narration-style annotation, e.g. *adjust the stove heat*. Though these type of labels are differentiable from a vision-side, they are near impossible to distinguish using body-worn sensors. To nevertheless give an estimate of activities present in the dataset, we provide activity verb counts (Ego4D and EPIC-Kitchens) and high-level activity counts (Ego-Exo4D) in Table 2.1. Our chosen sensor placement thus makes the CMU-MMAC [39], ADL [46] and ActionSense [42] dataset most comparable to the WEAR dataset. Compared to these three datasets, the WEAR dataset is the largest in size and participant count. Furthermore, the three existing benchmark datasets were all recorded in a non-changing indoor (kitchen) environment, which, in case of the ActionSense and CMU-MMAC dataset, was artificially created in a laboratory. Unlike the outdoor setup of the WEAR dataset, the indoor setup thus limits each dataset’s amount of variety captured in the visual data, as lighting conditions and surroundings remain the same throughout all participants. Lastly, as activities are mostly object-centric activities (e.g. cooking recipes) and action is mostly taking place in the POV of the user, prediction scenarios of the three datasets are more biased towards vision rather than inertial data.

2.2.2. Inertial-based HAR

Compared to video-based modalities body-worn sensor systems bear a great potential in analyzing our daily activities with minimal intrusion, yielding various applications from the provision of medical support to supporting complex work processes [26]. Within the last

Table 2.1.: List of available egocentric vision datasets, which provide inertial data, compared with the WEAR dataset. We differentiate between number (#Class) and type of activity classes (S = Sports, D = Daily Living, C = Cooking), number of participants (#Participants), recording environment (outside or inside), limb placement of IMU sensors (LW = left wrist, RW = right wrist, LA = left ankle, RA = right ankle) and type of recorded videos (T = trimmed or U = untrimmed video sequences). *Subset which provides both IMU and activity annotations **Exact participant counts are not provided; labels summarize key-steps † Activity verb classes. ‡ Excludes calibration periods.

Dataset	General				Where	Limb Placement				Video
	#Participants	#Class	#Hours	Type		LW	RW	LA	RA	
MEAD [154]	2	20	<1	D,S	In,Out					T
DataEgo [132]	≈10	20	4	D,S	In,Out					U
Ego4D [59]	≈60	87 [†]	160*	D,S,C	In,Out					T
Ego-Exo4D [60]	≤555**	689**	68*	D,S,C	In,Out					T
EPIC-Kitchens [38]	37	97 [†]	100	C	In					U
UESTC-MMEA-CL [192]	10	32	30	D	In,Out					T
CMU-MMAC [39]	13	16 [†]	2	C	In	✓	✓	✓	✓	U
ADL Dataset [46]	2	6	<1	D	In	✓	✓			U
ActionSense [42]	9	20	9 [‡]	C	In	✓	✓	✓	✓	U
WEAR	22	18	19	S	Out	✓	✓	✓	✓	U

decade deep learning based-methods have established themselves as the de facto standard in inertial-based HAR as they have shown to outperform classical machine learning algorithms [126, 65, 61]. One of the most well-known deep learning approaches for inertial-based HAR is the *DeepConvLSTM* which is a hybrid model combining both convolutional and recurrent layers [126]. By combining both types of layers the network is able to automatically extract discriminative features and model temporal dependencies. Following the success of the original DeepConvLSTM, researchers worked on extending the architecture [121, 188] or build up on the idea of combining convolutional and recurrent layers by proposing their own architectures [191, 1, 200, 210]. Within this publication we are reporting benchmark scores using the WEAR dataset inertial sensor-streams as input for two popular HAR models [17, 1]. Contrary to the belief that one needs to employ multiple recurrent layers when dealing with sequential data [89], Bock et al. [17] proposed an altered *shallow DeepConvLSTM* architecture which proved to outperform the original architecture by a significant margin. Differently, Abedin et al. [1] chose to build up on the idea of the DeepConvLSTM and introduced the *Attend-and-Discriminate* architecture which exploits interactions among different sensor modalities by introducing self-attention through a cross-channel interaction encoder and adding attention to the recurrent parts of the network.

2.2.3. Vision-based HAR

Predicting activities performed by humans based on visual-cues can broadly be categorized into three main application scenarios: action recognition, localization and anticipation. Action recognition systems [112, 180, 102] aim to assign a set of trimmed action segments an activity label. Contrarily, TAL systems [204, 195, 111] are tasked to identify start and end times of all activities in a untrimmed video by predicting a set of activity triplets (*start*, *end*, *activity label*). Lastly, action anticipation systems [56, 144] aim to predict the label of a future activity having observed a segment preceding its occurrence. Though sensor-based HAR systems are employed using a sliding window approach and thus assign activity labels to a set of trimmed inertial-sequences, their ultimate goal is to identify a set of activities within a continuous timeline. We therefore deem vision-based TAL to be most comparable to inertial-based HAR and will focus on it in our benchmark analysis. Existing TAL methods can be divided into two categories: two- and single-stage approaches. Two-stage approaches [106, 103, 193, 9, 208, 202, 58, 110, 134, 156, 212, 207, 165] divide the process of TAL into two subtasks. First, during the action segment proposal generation, candidate video segments are generated which are then, classified with an activity label as well as refined regarding their temporal boundaries. Contrarily, single-stage approaches [195, 152, 122, 111, 107, 114, 104, 30, 204, 151] aim to localize actions in a single shot without using action proposals.

In light with the success of transformer architectures in natural language processing [177, 44] and computer vision [93, 199, 112], researchers have demonstrated their applicability for TAL [36, 109, 111, 152, 165, 204] breaking previously held benchmark scores of numerous popular datasets [71, 38, 88] without any additional training data by a significant margin. One of such architectures is the *ActionFormer* proposed by Zhang et al. [204], which is an end-to-end trainable transformer-based architecture, which unlike other single-stage approaches, does not rely on pre-defined anchor windows. The architecture combines multiscale feature representations with local self-attention and is trained through a classification and regression loss calculated by a light-weighted decoder. Building up on ActionFormer architecture, Shi et al. [151] proposed the *TriDet* model which suggest to replace the transformer layers of the ActionFormer with fully-convolutional, so-called SGP layers, as well as use a trident regression head which claims to improve imprecise boundary predictions via an estimated relative probability distribution around the boundary. Given the rapid rise in popularity of single-stage TAL such as the ActionFormer, we decided said models to be a suited option to deliver a first benchmark for the WEAR dataset.

2.2.4. Multimodal (Inertial and RGB Video) HAR

With early works such that of Spriggs et al. [155] having shown the complementarity of inertial- and camera-based features, research has followed up by exploring different ways of combining the two modalities. One can categorize such methods broadly by the point in time at which the fusion of both modalities is performed. Late fusion approaches usually follow a two-stream architecture training both vision- and inertial-based modalities separately before merging together outputs of each stream through such as produced softmax probabilities e.g. via a weighted combination [184], attention mechanisms [54], pooling operations [153, 82], majority voting [47], a concurrent classifier [187, 45, 80, 161] or knowledge transfer between separate networks [136]. Early fusion approaches aim at jointly learning from both modalities by using feature embeddings calculated on one (or both) modalities to e.g. use the concatenation of both to train a concurrent network [81, 192, 124, 115, 79, 48, 45, 46, 154, 197, 29, 85, 84], enhance softmax probabilities used during late fusion [45, 46] or adding intermediate cross-view connections amongst the two modality streams [80]. With experiments showing that single-stage TAL models are able to produce competitive results on raw inertial data, this paper also tests the applicability of two state-of-the-art models, namely the ActionFormer and TriDet model, to fuse and combine cues of both modalities in an early-fusion style. Unlike other early fusion techniques, our approach is the first to directly use the raw inertial data by means of simple concatenation together with a vision-based feature embedding.

2.3. Methodology

2.3.1. Study Design & Scalable Pipeline

Participants of the WEAR dataset were recorded during separate recording sessions. Prior to their first session, participants were handed a recording plan which outlined the study protocol as well informed about any risks of harm, data collection, usage, anonymisation and publication, as well as how to revoke their data usage rights at any point in the future. The study design involving human participants was reviewed and approved by the University of Siegen (reference number: 03/2023 VVT). All participants were briefed and provided their written informed consent. Each participant was asked to perform all 18 workout activities detailed in the recording plan. The location and the time of day at which the sessions were performed, were not fixed and thus vary across participants (see Section A.2 in the Appendix). Participants were suggested to follow a two-session setup, i.e. 9 activities per session. Nevertheless, it was allowed to differ from this setup and split the 18 activities across as many (or as few) sessions as participants liked. This caused the amount of recording

sessions to vary across participants, but also increased the amount of captured variability in weather conditions and recording locations. In order to avoid misunderstandings in the execution of the activities, the authors discussed all activities prior to each session and encouraged participants to ask questions during the session if something remained unclear. Participants were tasked to perform each activity for roughly 90 seconds. As activities varied in their intensity, it was not required to perform activities for 90 seconds straight and participants could include breaks as needed. Furthermore, to ensure that each participant was able to perform all workout activities properly, the recording plan detailed how activities could be altered in their execution, for instance so that they required less physical strength.

2.3.2. Experimental Protocol

In order to properly explain to participants the activities they needed to perform and give insights on the overall study design a recording plan (see Section A.4 in the Appendix) was provided to participants prior to their first session. The recording plan details all necessary materials and is written in such a way that the can easily be reproduced by persons other than the authors. The plan further outlines the study protocol as well informs about any risks of harm, data collection, usage, anonymisation and publication, as well as how to revoke data usage rights at any point in the future. Besides a written description of each activity, the original document provides short video-clips of each activity, showing the correct execution of exercises. To avoid any misunderstandings, the participants further received a one-on-one session with the researchers being able to ask their questions about the plan and activities in it.

The recording plan provided with our dataset includes all necessary materials and is written in such a way that all activities and sessions can easily be reproduced by persons other than the authors. Besides the used sensors for video and acceleration recording, the exercises only require a yoga mat and a chair (or similar items). Sessions can be recorded at any location outside as long as the privacy of the participants and bystanders is ensured. The code repository of the dataset provides a detailed guide on how to work with the equipment and how to collect, postprocess and annotate newly collected data. We argue that this facilitates reproducibility, and with a minimal setup ensures that it is possible for others to extend our dataset at a later date.

2.3.3. Participant Information

We recorded data for 22 participants (13 male, 9 female) at 11 different locations and under varying weather conditions. The first 18 participants were recorded over a stretch of 5 months (October to February), totalling more than 15 hours. In an effort to provide a test set along

with the WEAR dataset an additional 6 participants were recorded, totalling around 4 hours of additional data. To allow researchers to explore also personalised prediction approaches, the test dataset includes two participants which were already part of the first recordings, and which volunteered to complete the workout an additional time. Recordings of the test dataset took place during spring (new participants) as well as summer (re-recordings). Egocentric video data of the four new participants was recorded using a GoPro Hero 11 as opposed to a Hero 8 and two new participants were recorded at a previously unseen location. On average each participant contributed roughly 50 minutes of data. At the time of recording, the participants had a mean age of 27.59 years (standard deviation (SD): 4.69), a mean height of 175.2 cm (SD: 9.83), and a mean weight of 69.74 kg (SD: 11.23). In order to assess their sports level, participants filled in a post-session questionnaire. The questionnaire contained questions related to vital information (such as body height, weight and age), weekly workout frequency (min. 15 minutes duration) and experience in particular workout activities. The participants in the study worked out for a mean of 3.73 times per week (SD: 2.31), were familiar with a mean of 15 out of the 18 activities in advance (SD: 3.57), and regularly performed a mean of 5.71 of the recorded activities (SD: 4.14) as part of their private workouts. Participants reported for their personal workout schedules a wide-range of cardio- (running, hiking, cycling, dancing), strength- (weight lifting, freeletics, rowing, bouldering), team- (volleyball, basketball, (table-)tennis) and flexibility-focused (yoga, ballet) exercise types. Individual answers of each participant can be found in the Appendix (see Section A.2).

2.3.4. Dataset Collection & Structure

The WEAR dataset provides participant-wise raw and processed acceleration and egocentric-video data (see Figure 2.1). We focus on 3D accelerometers especially as they cover a substantial amount of commercial fitness devices worn at the wrists and ankles. They furthermore are used in a large set of existing research and datasets focusing on wearable data for activity recognition, and they do not suffer from noise, drift, and other device-specific characteristics. 3D accelerometer data was collected at 50 Hz with a sensitivity of $\pm 8g$ using four open-source Bangle.js smartwatches running a custom, open-source firmware [175]. The watches were placed by the researchers in a fixed orientation on the left and right wrists and ankles of each participant. Egocentric video data was captured using either a GoPro Hero 8 or Hero 11 action camera, which was mounted using a head strap on each participant’s head. The resulting ‘.mp4’-videos were recorded at 1080p resolution with 60 frames per second and the camera being tilted downwards in a 45 degree angle. A second tripod-mounted camera was placed within the proximity of each participant to facilitate annotation recording the environment in which the workout was performed from a third-person-perspective. Using

again a large FOV setting, the second camera was placed in a way such that as much area as possible was captured. To allow for even more freedom of movement, participants were allowed to move out of the FOV of the second camera, but were asked to start and end their activities within the camera’s FOV. This allowed participants, especially during running exercises, to run straight distances and overall commence activities in a more natural way. For privacy reasons, the second camera’s video and all audio captured are not part of the WEAR dataset.

The open-source firmware [175] running on each Bangle.js smartwatch stores the lossless, delta-compressed inertial data in separate files on the internal memory of each watch. During post-processing, said compressed files were extracted, uncompressed and concatenated to a single ‘.csv’-file per session. Being a common issue with accelerometers sampling at a high sampling rate, the Bangle.js smartwatch is not able to maintain an exact sampling rate of 50 Hz at all times throughout the experiment, with the true sampling rate being closer to 48 Hz with fluctuations ranging between ± 1 Hz. The firmware provides for each file a timestamp that was set by the on-board real-time clock, which allows correcting individual times of all delta-compressed samples. Therefore, in order to obtain the true sampling rate and correct the timestamps of the concatenated ‘.csv’-file, synchronisation jumps were performed by each participant at the start and end of each session. The synchronization jumps involved participants move in front of the tripod-mounted camera, stand still for approximately 10 seconds, jump three times along with raising the arms while jumping, and stand still for another 10 seconds. This allowed to map peaks in the inertial sensor streams to be mapped to points in the video stream and thus obtain a start and end point within both modality data streams. Lastly, assuming recorded inertial data records are equidistant, all records within the span of the start and end-point were evenly distributed across the experiment’s duration and, as a final step, resampled to have a sampling rate of 50 Hz via linear interpolation. Similar to the inertial data, the video data recorded by the head-mounted GoPro was not recording at a true frame rate of 60 FPS, but slightly deviated from that (i.e. 59.94 FPS). We therefore also resampled the egocentric videos to be of a frame rate of 60 FPS.

In order to validate our synchronization process we made use of the similarities between sensor and audio data and converted each axis of the 3D accelerometer as well as their combined magnitude to four separate ‘.wav’-files. This approach is inspired by the works of Scholl et al. [148] and Bin Morshed et al. [13]. We calculated the magnitude as the summed norm of each individual inertial sensor channels, i.e. $\sqrt{x^2 + y^2 + z^2}$ with x , y and z being the x-, y- and z-axis of the 3D accelerometer data. Having converted the ‘.csv’-data to ‘.wav’-files allowed us to import both video data and inertial data into a standard video editing tool, in our case we used Final Cut Pro (see Figure 2.2). The user interface of Final Cut offers to see previews of sound files being in our case equivalent to a graph-like visualization of the

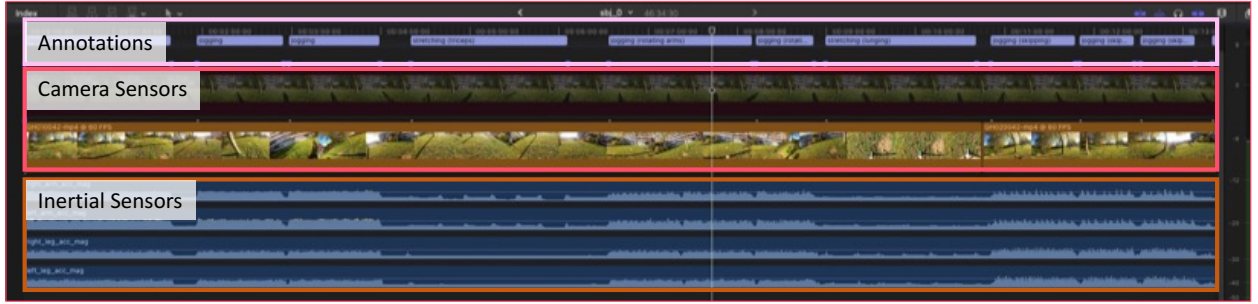


Figure 2.2.: Snapshot along with descriptions of the annotation process using Final Cut Pro. Importing the converted video and inertial data (as ‘.wav’-files) allowed for an easy validation of the synchronization process. Labels were added via subtitles, exported as ‘.srt’-files and converted such that they can be appended to the respective ‘.csv’-files.

acceleration data. This allowed us to verify our applied synchronization during annotation throughout the whole duration of each participant’s session. On average, the combined magnitude proved to be most useful when verifying the correctness of our synchronization across time. Labels of the activities were added by a single expert annotator as subtitles in ‘.srt’-format. A final script then converted the exported ‘.srt’-file to ‘.csv’-format, filling gaps within the subtitles with a *NULL* label and appended this to the respective final inertial sensor data ‘.csv’-file.

2.4. Benchmarks and Baseline Results

Though the WEAR dataset provides the possibility for a multitude of HAR use cases, this paper focuses on introducing one sample application scenario per data modality, namely: (1) inertial-based wearable activity recognition, (2) vision-based TAL, as well as, (3) a combined approach using both data modalities as input simultaneously. We chose to use said application scenarios because of their similarities with each other as they both aim to detect a set of activities in an untrimmed sequence of data. Nevertheless, other HAR-specific (e.g. action anticipation and classification) and non-HAR application scenarios (e.g. hand detection, pose estimation or simultaneous localization and mapping (SLAM)) are applicable.

During each experiment we employ a Leave-One-Subject-Out (LOSO) cross-validation which has each participant being used as the validation set exactly one time while all other participants are used as training data. In order to minimize the risk of performance differences between experiments being the result of statistical variance, all experiments mentioned in this publication were repeated three times employing three different random seeds (1, 2 and 3). The final validation results of an experiment are then reported as the average across the three repeated runs. As mentioned in Chapter 2.3.1 a separate test dataset was collected

which consists of 6 additional participants performing the same workout as all the other participants. The test dataset further includes additional sessions of two participants which volunteered to be recorded a second time. We use the test data to get an unbiased assessment of our chosen set of hyperparameters and applied postprocessing, which are determined during the single-modality and multimodal validation experiments.

Note that all models part of our benchmark analysis were applied without pretraining on existing benchmark datasets. Nevertheless, feature embeddings to train the vision models were extracted using a two-stream I3D feature extractor [27] which was pre-trained on Kinetics-400 [90]. With the standard error of evaluation metrics amongst runs being at maximum 2.5% and the majority of runs being below 1%, we only report average evaluation metrics in this paper. All mentioned experiments were conducted on a single NVIDIA Tesla V100 GPU and lasted no longer than 24 hours. Though sharing inherent similarities, vision-based action localization algorithms predict a collection of activity segments defined by a start and end time, while, contrarily, inertial-based HAR systems provide labels based on the pre-defined windowed segmentation. Given their difference in prediction output, different evaluation metrics are applied, with mAP being most prominent metric in vision-based TAL and F1-score being the most prominent metric in inertial-based activity recognition. Therefore, to guarantee comparability amongst application scenarios and architectures, predictions of each algorithm are converted such that both vision- and inertial-based evaluation metrics can be calculated. More specifically, our reported benchmark evaluation metrics are (1) a record-based calculated recall, precision and F1-score, and (2) segment-based mAP at different temporal Intersection over Union (tIoU) thresholds, commonly used to evaluate TAL datasets. To ensure readability we only provide a selection of visualized results. For a complete collection visualizations of all mentioned experiments, please see Chapter C.7 within the supplementary material.

2.4.1. Single-Stage Temporal Action Localization for Inertial Data

Though originally intended to be applied to video data, our work demonstrates that vision-based TAL models, such as the TriDet [151] and ActionFormer [204] models, prove to be applicable to inertial data as well as a means to fuse both modalities. Both TAL architectures take as input a collection of clip-wise feature embeddings, which are obtained by applying a sliding window approach on top of an input video. Using both classification and regression losses, the ActionFormer and TriDet model then try to localize activity segments, defined by an activity label, start, and end time, within a complete video. To obtain discriminative feature embeddings of each sliding video clip, the TAL community has resorted to using feature extraction methods such as a two-stream I3D feature extractor [27]. The feature

extractors, usually pretrained on a larger vision corpus like the Kinetics-400 dataset [90], summarize the raw visual data into a one-dimensional feature embedding.

As both TAL and inertial-based architectures rely on a sliding window approach to preprocess data, our paper suggests a simple yet effective preprocessing technique such that raw inertial data can be used to train vision-based TAL models. Illustrated in Figure 2.3, we propose vectorizing the two-dimensional sliding window data commonly found in inertial-based architectures, creating one-dimensional raw inertial feature embeddings per sliding window that can be used to train TAL models such as ActionFormer and TriDet. Specifically, starting with the windowed inertial data of dimensions [*no. windows*, *window length*, *no. sensor axes*], we concatenate the individual sensor axes of each window, resulting in a one-dimensional feature vector of size [*window length* $\times$ *no. sensor axes*]. In the case of the WEAR dataset, which provides 12 individual sensor axes per participant, we obtain (depending on the window length) feature vectors of size 300 (0.5 seconds), 600 (1 second), and 1200 (2 seconds) per video clip, i.e., sliding window. Even though our concatenation approach results in varying input dimensions, this change does not come at increased computational costs. More specifically, while the number of learnable parameters marginally increases (not more than 10%) with an increased input dimension, unlike other approaches, no additional embedding needs to be extracted from the inertial data, and raw data streams can be directly used. Furthermore, as demonstrated in the multimodal experiments in this paper, we show that by concatenating one-dimensional inertial data and I3D feature embeddings, the TAL architectures are capable of successfully fusing multimodal information, significantly outperforming single-modality approaches.

By using our approach, the TAL architectures process the inertial data in a different format compared to, for example, the shallow DeepConvLSTM. However, their performed feature extraction is comparable. Specifically, both architectures shift convolutional filters separately across each sensor axis, with the only difference being that the TAL models do not pad the start and end of each sensor axis, allowing for a slight overlap amongst axes at their point of concatenation. Nevertheless, we argue that our approach draws inspiration from visual feature embeddings such as the I3D features, which have proven effective in training architectures, yet, by design, will also cause an overlap when transitioning between flow and RGB features.

2.4.2. Single-Modality Experiments

Similar to Zhang et al. [204] and Shi et al. [151], we opted to train the vision-based benchmark models using two-stream I3D feature embeddings with three different clip lengths (0.5, 1, and 2 seconds), employing a 50% overlap between clips. Besides extending the number of

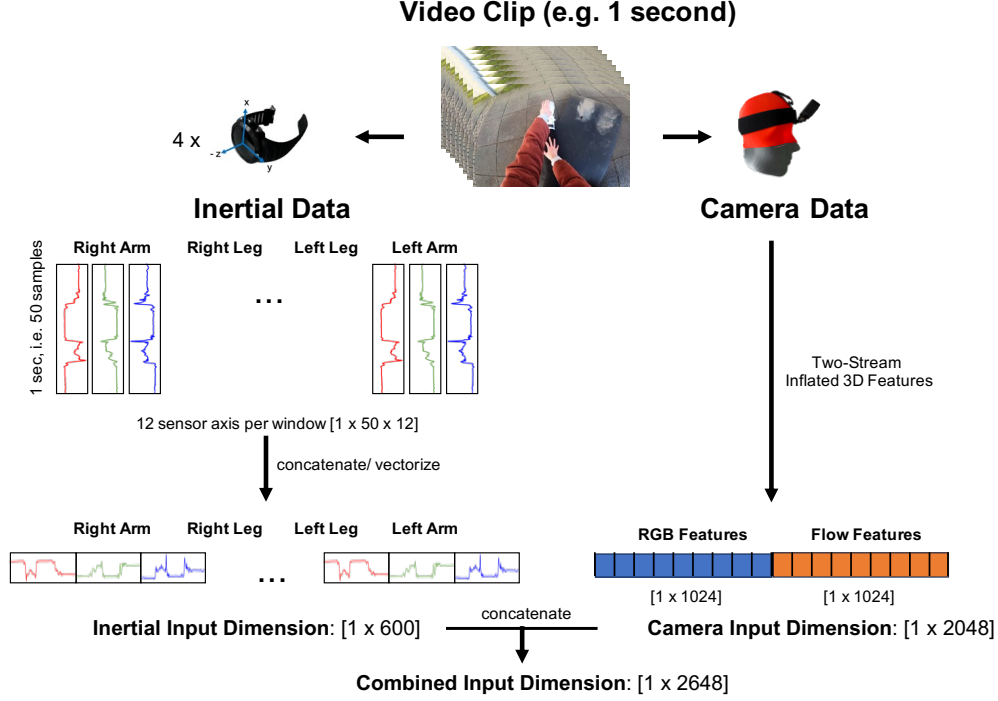


Figure 2.3.: Visualization of the applied preprocessing on inertial and camera data in order to make to create a feature embedding which can be used to train the TriDet and ActionFormer network.

epochs to 100, we adopted the same training strategy that yielded the best results on the EPIC-Kitchens dataset [38], as reported by both architectures. In contrast to inertial-based approaches, TAL models are not trained to predict an explicitly modeled NULL-class. With both models set to predict up to 2000 action segments per video, each timestamp ended up being classified by an action segment, resulting in a prediction performance for the NULL-class close to or at 0% accuracy. To address this, we eliminated low-scoring segments by increasing the scoring threshold of both models to 0.1, significantly enhancing the accuracy of the NULL-class while only marginally affecting prediction performance of all other activity classes.

As our inertial-based benchmark algorithms of choice, we employed the shallow DeepConvLSTM proposed by Bock et al. [17] and the Attend-and-Discriminate model proposed by Abedin et al. [1]. Throughout all experiments, we followed the training strategy suggested by Bock et al. [17], which has shown to yield reliable results on various inertial-based HAR datasets, only extending the number of epochs to match those of the vision-based experiments. To compensate for longer training times, we applied a step-wise learning rate schedule. Additionally, incorporating architecture changes recommended by Bock et al. [17], we modified the Attend-and-Discriminate model to utilize a one-layered instead of a two-layered

recurrent module and adjusted the convolutional kernel size based on the sliding window and sampling rate of the WEAR dataset. Given that inertial-based architectures provide predictions on a per-window basis, frequent intermediate, short-lasting activity switches occur along the time axis, resulting in small coherent segments and ultimately lower mAP scores compared to the vision-based models presented in this paper. To mitigate these switches, predictions made by the inertial-based architectures were smoothed using a majority-vote filter of 10 seconds. For detailed ablation results on the design choices and chosen hyperparameters see Section C of the supplementary material.

Examining the results presented in Table 2.2, one can observe that for the TAL models, a clip length of 1 second yielded the best predictive performance, while for the inertial-based architectures, a larger window size of 2 seconds produced higher classification results. Analyzing per-class results, it is evident that when using only visual data as input, the models struggle to differentiate between different running styles, activities outside the field of view of the participant (e.g., triceps stretches), as well as normal and complex sit-ups. In contrast, since inertial sensors are not affected by occlusions and accelerometer data provides an encoded representation of limb orientation due to the Earth’s gravitational force, algorithms trained on inertial data do not exhibit these weaknesses and are particularly reliable during activities where limb orientation is the main discriminator (e.g., stretches). However, as indicated by a larger NULL-class accuracy compared to the inertial-based models, vision-based approaches are better at differentiating activities from breaks, which is a desirable trait of HAR systems given the often substantial size of the NULL-class [26].

Interestingly, using the same hyperparameters as those used during vision-based experiments, a plain ActionFormer and TriDet network not only produce competitive classification results based on inertial input data but also show less confusion among activity classes compared to inertial-based architectures. Comparing per-class results between the TAL models and classic inertial-based architectures, it is evident that the confusion in the TAL models mostly lies within the activity categories themselves (jogging, stretching, and strength), whereas inertial-based models exhibit a higher degree of overall confusion between the activities and the NULL-class. Given that inertial-based models like the DeepConvLSTM are trained to classify sliding windows individually, we assume that this overall larger confusion and intermediate activity switches are caused by the inertial models’ inability to leverage context information beyond the window they are tasked to classify. Consequently, due to the intermediate activity switches, calculated mAP scores of the inertial architectures are significantly lower than those of the camera-based approaches. Nonetheless, inertial-based models are on average able to predict all workout activities more consistently and produce the highest classification metrics across all experiments.

2.4.3. Multimodal (Inertial and Egocentric Video) Experiments

As demonstrated in previous chapters, both the vision and inertial modalities present various strengths and weaknesses. Therefore, in an effort to leverage the strengths of both modalities, we evaluated a combined, multimodal training approach using the TriDet and ActionFormer model. To fuse the two-stream I3D feature embeddings with the inertial data early on, we vectorized the inertial data, as described in the previous section, such that each sliding window is represented by a one-dimensional inertial embedding vector. We then concatenated said vector with the I3D feature embedding of the corresponding video clip, resulting in an early fused representation of the a sliding window consisting of both visual and inertial information. Through simple concatenation of both modalities, both architectures achieved the highest average mAP and close-to-best F1-scores across all experiments (see Table 2.2). Comparing confusion matrices of all three approaches (see Figure 2.4) reveals that both vision models, applied in a plain fashion, are able to successfully combine inertial and vision data and leveraging the previously mentioned strengths of each modality. To assess how our early-fusion approach compares to voting-based late-fusion approaches such as proposed by Ijaz et al. [80], we implemented an *Oracle*-based late fusion, which creates perfectly late fused predictions of different models. The predictions are merged by comparing each of them with the ground truth data and only keeping, if predicted by one of the networks, the correct prediction. Interestingly, the first *Oracle*-late-fusion $O-LF(I, C)$, which late fuses predictions of the best inertial and best vision model, produces lower mAP scores than that of the best TAL model being trained on both modalities simultaneously. Furthermore, late-fusing the best inertial, vision and early-fusion approach ($O(I, C, I + C)$), increases classification and mAP scores of $O(I, C)$ by as much as 10%, suggesting the early-fusion-based approach is capable of learning to differentiate activities both single-modality models failed to classify correctly (see e.g. misclassified stretching exercises in Figure 2.5). Nevertheless, classification results of the *Oracle*-based late fusion significantly outperform both single- and combined-modality approaches, indicating that the data set is far from being saturated.

2.4.4. Test Set Results

To verify chosen hyperparameters and suitability of the applied postprocessing as determined in the LOSO cross-validation, we collected an additional test set of six additional participants. The participants constitute of four previously unseen participants as well as two participants, already present in the training data and which volunteered to be recorded a second time. Unlike recordings of the training data, the test data was recorded during spring and summer. Egocentric video data of the four new participants was recorded us-

Table 2.2.: LOSO validation results of human activity recognition approaches based on body-worn IMU (Inertial), vision (Camera) and combined (Inertial + Camera) features for different clip lengths (CL) on the first 18 participants of the WEAR dataset evaluated in terms of precision (P), recall (R), F1-score and mean average precision (mAP) for different temporal intersection over union (tIoU) thresholds. The results underline the complementarity of the inertial and camera modalities. *O-LF()* corresponds to the *Oracle*-based late fusion, which creates perfectly late fused predictions of different models. Best results per modality are in **bold**.

	Model	CL	P	R	F1	mAP					
						0.3	0.4	0.5	0.6	0.7	Avg
Inertial	Shallow D.	0.5s	74.81	74.93	72.26	60.03	57.91	55.54	54.19	52.31	55.99
	A-and-D	0.5s	76.54	73.04	72.10	57.51	54.59	52.22	49.47	46.94	52.15
	ActionFormer	0.5s	63.09	78.17	66.80	81.12	77.47	69.08	53.98	38.56	64.04
	TriDet	0.5s	68.58	78.30	70.41	79.66	76.07	69.78	60.57	49.95	67.21
	Shallow D.	1s	77.71	77.41	75.44	63.37	61.35	58.91	57.02	55.05	59.14
	A-and-D	1s	79.97	75.92	74.67	61.72	58.77	56.29	53.83	51.59	56.44
	ActionFormer	1s	65.88	78.44	68.40	81.91	79.75	76.99	72.09	64.33	75.01
	TriDet	1s	68.18	78.58	70.36	81.89	80.46	78.47	74.59	68.91	76.86
	Shallow D.	2s	78.00	77.19	75.43	65.14	63.58	60.98	59.15	57.39	61.25
	A-and-D	2s	80.96	78.42	77.10	63.52	61.57	58.82	56.88	54.41	59.04
	ActionFormer	2s	61.56	76.63	64.57	80.14	77.24	74.41	70.24	63.18	73.04
	TriDet	2s	63.14	75.64	65.94	78.69	77.02	73.89	70.68	65.31	73.12
Camera	ActionFormer	0.5s	60.86	72.98	62.54	81.07	78.48	72.13	57.34	41.27	66.06
	TriDet	0.5s	64.69	72.89	64.84	80.46	76.95	72.62	64.46	55.52	70.00
	ActionFormer	1s	65.82	75.34	66.40	85.19	83.31	80.94	77.22	69.96	79.32
	TriDet	1s	67.42	75.21	66.88	84.88	83.49	82.12	79.75	76.24	81.30
	ActionFormer	2s	63.40	76.25	65.39	84.93	82.95	80.58	77.55	72.07	79.62
	TriDet	2s	65.53	75.36	65.91	82.94	82.00	80.22	78.13	75.20	79.70
Inertial + Camera	ActionFormer	0.5s	71.45	83.77	74.51	86.04	83.96	77.80	64.41	43.76	71.19
	TriDet	0.5s	75.74	82.79	76.83	84.45	82.07	77.12	68.81	57.16	73.92
	ActionFormer	1s	72.87	83.24	75.26	86.82	85.27	82.46	78.26	72.33	81.03
	TriDet	1s	73.39	82.00	75.09	86.42	85.48	83.35	80.39	75.64	82.26
	ActionFormer	2s	65.86	80.17	68.82	84.18	81.66	78.17	74.59	68.74	77.47
	TriDet	2s	69.19	81.09	71.98	84.66	82.79	80.66	78.11	74.15	80.07
	<i>O-LF(I, C)</i>	<i>0.5s</i>	<i>90.23</i>	<i>91.52</i>	<i>89.60</i>	<i>74.79</i>	<i>73.73</i>	<i>72.44</i>	<i>69.94</i>	<i>68.53</i>	<i>71.88</i>
	<i>O-LF(I, C)</i>	<i>1s</i>	<i>91.26</i>	<i>92.72</i>	<i>90.68</i>	<i>75.37</i>	<i>74.42</i>	<i>73.64</i>	<i>72.74</i>	<i>71.41</i>	<i>73.52</i>
	<i>O-LF(I, C)</i>	<i>2s</i>	<i>91.26</i>	<i>92.94</i>	<i>90.81</i>	<i>73.46</i>	<i>72.37</i>	<i>71.70</i>	<i>70.50</i>	<i>69.60</i>	<i>71.53</i>
	<i>O-LF(I, C, I + C)</i>	<i>0.5s</i>	<i>93.29</i>	<i>94.37</i>	<i>93.10</i>	<i>84.41</i>	<i>84.17</i>	<i>83.37</i>	<i>81.61</i>	<i>80.40</i>	<i>82.79</i>
	<i>O-LF(I, C, I + C)</i>	<i>1s</i>	<i>93.99</i>	<i>94.89</i>	<i>93.38</i>	<i>84.14</i>	<i>83.81</i>	<i>83.34</i>	<i>82.54</i>	<i>81.81</i>	<i>83.13</i>
	<i>O-LF(I, C, I + C)</i>	<i>2s</i>	<i>93.69</i>	<i>95.13</i>	<i>93.38</i>	<i>81.79</i>	<i>81.37</i>	<i>81.02</i>	<i>79.90</i>	<i>78.94</i>	<i>80.60</i>

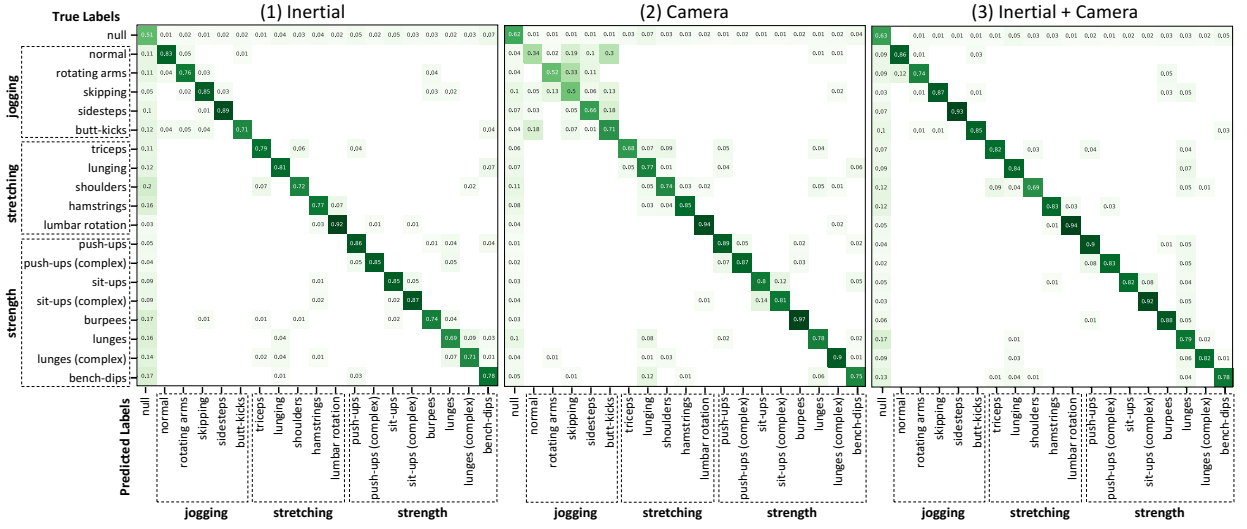


Figure 2.4.: Confusion matrices of the TriDet model [151] being applied using inertial, vision (camera) and both combined (inertial + camera) with a one second sliding window and 50% overlap. Compared to inertial-based architectures [17, 1] overall confusion (except for the NULL-class) is decreased. After combination strengths of each architecture are leveraged with e.g. jogging activities not getting confused anymore and overall confusion with the NULL-class decreases. Note that confusions which are 0 are omitted.

ing a GoPro Hero 11 as opposed to a Hero 8 and two new participants were recorded at a previously unseen location. Figure 2.6 summarizes results of the benchmark algorithms applied on the test dataset using a 1 second sliding window with a 50% overlap. Training was performed identical to that described in previous chapters with the input data being the data of the 18 participants used for LOSO validation. Each experiment was repeated three times with different random seeds (i.e., 1, 2 or 3). One can see that observed trends are similar to those seen during LOSO cross-validation. Nevertheless, one can observe a increase in average mAP and classification metrics for both the inertial- and vision-based models, which does also translate to mAP and class increasing when combining camera and inertial data.

2.5. Ablation Experiments

2.5.1. Selection of Inertial Sensors

Although commercial fitness products worn on the ankle or foot (e.g., anklets, sock-embedded devices, or shoe-embedded units) have gained popularity in recent years, wrist-worn sensors like smartwatches remain the most popular body position for inertial sensors in the context of wearable activity recognition. The following experiments (see Figure 2.7) investigate how reported LOSO results in Chapter 2.4.2 and 2.4.3 are influenced by using only a subset of

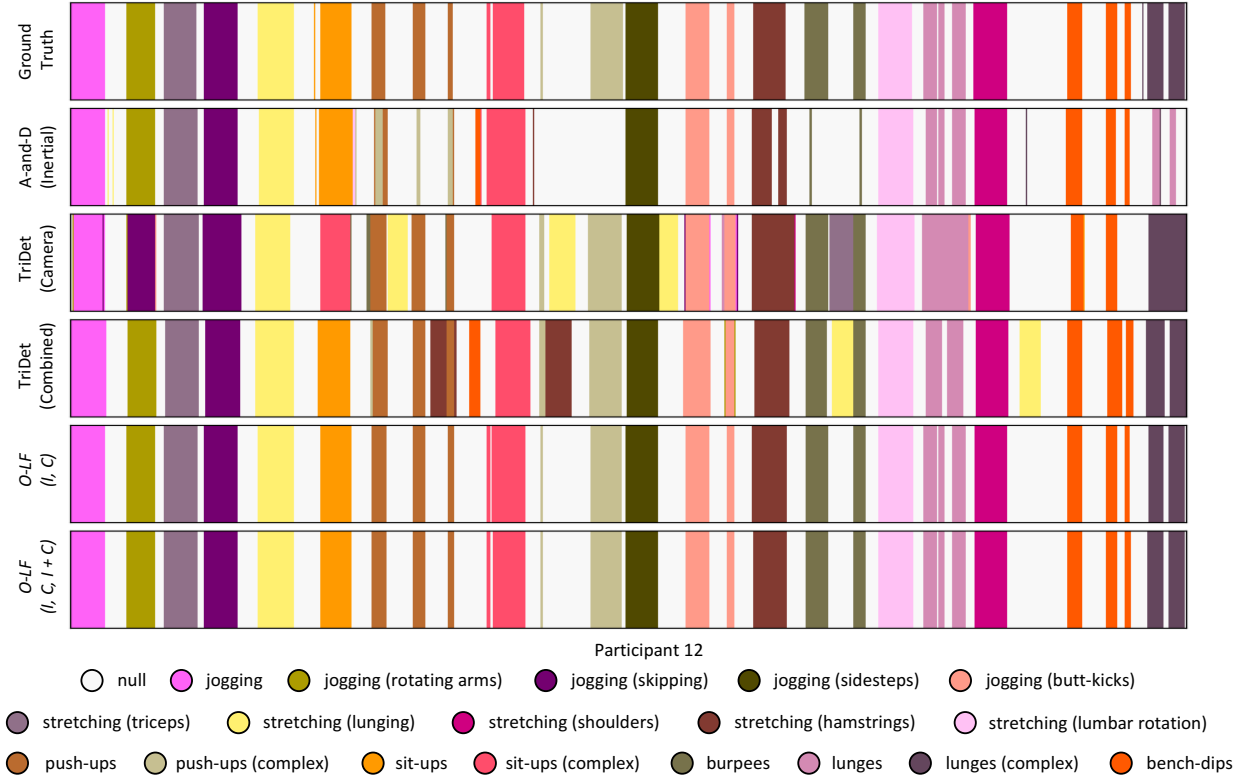


Figure 2.5.: Color-coded comparison of the ground truth data of a sample participant with the best inertial-based (A-and-D), camera-based (TriDet) and fusion-based model (TriDet) along with an oracle combination of the best fusion-based model ($O-LF(I, C)$) as well as an oracle combination the best camera, inertial and fusion-based-model ($O-LF(I, C, I + C)$) using a sliding window approach of 1.0 seconds with a 50% overlap. The visualisation underlines the similarities amongst the predictive streams of $O-LF(I, C)$ and the fusion-approach as well advantages of learning from both modalities simultaneously.

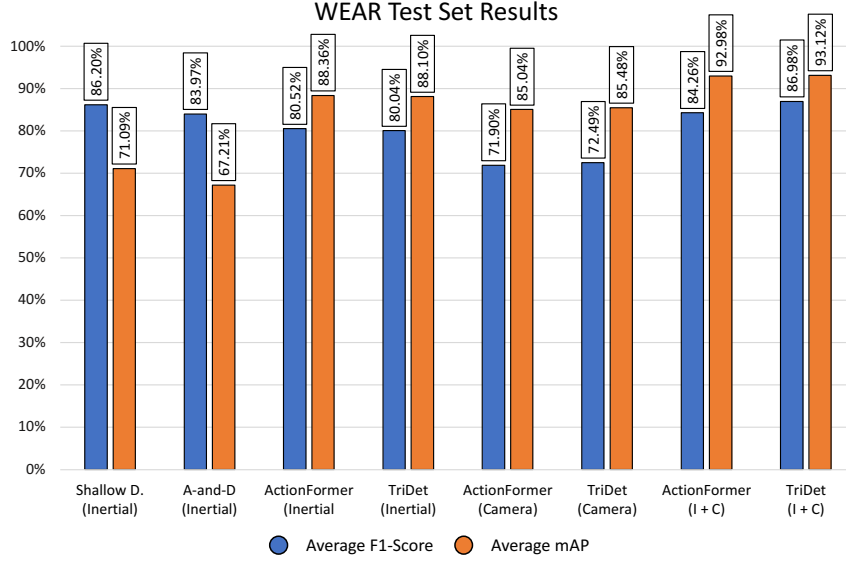


Figure 2.6.: Average F1-score and mAP on the WEAR test set. The test features 4 unseen participants as well 2 reoccurring ones, a unseen location, different weather conditions and a new camera sensor. One can see that observed trends are similar to those seen during LOSO cross-validation.

the inertial sensors, specifically by using only (1) acceleration recorded from the right wrist and (2) acceleration recorded from both the right wrist and right ankle. Results show that using only acceleration data obtained from the right wrist significantly decreases predictive performance across all algorithms and metrics. Moreover, the value of additionally measuring acceleration at the ankles of participants is clearly underlined, as results again significantly increase, mostly on par compared to using all four inertial sensor locations. Interestingly, unlike the inertial-based architectures, results of the vision-based TAL models improve when excluding data captured by the left wrist and left ankle inertial sensors, which could be due to the dataset being biased towards right-handed participants (see Table 3 in the supplementary material) and dominant hand movement being overall more consistent.

2.5.2. Second Execution of Workout Sessions

As mentioned in Chapter 2.3.1, as part of the test dataset provided along the WEAR dataset, we recorded all activities of two participants (sbj_0 and sbj_14) a second time in August. Both participants recording conditions significantly differed from their first recording, with temperatures being around 25 degrees Celsius with overall more sunny weather conditions. Further, as not all participants knew all activities beforehand (see Table A.3 in the Appendix), recording the same participants a second time would allow to analyse how a certain degree of familiarity with the recording setup can be seen in altered movements (e.g., via a

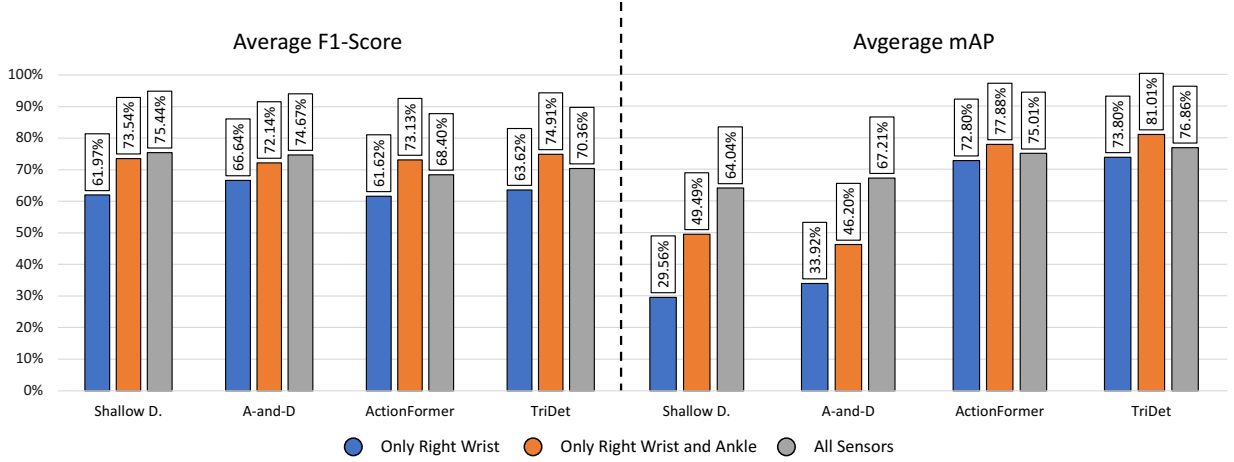


Figure 2.7.: Results using only inertial sensors placed on the right wrist (RW) and right wrist + ankle (RW + RA) compared with using all sensors for a clip length of 1 second on our WEAR dataset evaluated in terms of F1-score and mean average precision (mAP) averaged across different temporal intersection over union (tIoU) thresholds (0.3:0.7:0.1). Using only wrist-worn data can see a clear overall decrease across all evaluation metrics.

smoother execution of activities) as well as participant-specific finetuning affects the overall recognition performance. Figure 2.8 compares validation results obtained on the first recording of sbj_0 and sbj_14 with their second execution of the workout plan in August. Unlike our prior experiments, each algorithm is trained using the validation data as reported in Chapter 2.4.2 and 2.4.3 except the participant which is to be analysed. All results are postprocessed as reported Chapter 2.4.2 and 2.4.3. While, one can see improved results regarding sbj_0, which only knew five of the workout activities prior to participating in the study, this trend does not apply to sbj_14. More specifically, improvements and decline rates between the two recordings lie within the expected standard deviation across participants (between 15% to 20%). Though being a small sample size of only two participants, the results suggest that in order to guarantee a reliable detection of activities, each participant would need to be recorded multiple times under different conditions. Nevertheless, in order to come up with reliable conclusions, future extensions of the WEAR dataset would need focus on re-recording more participants multiple times under varying conditions.

2.6. Limitations

Our dataset contributes a benchmark for human activity recognition classifiers, for the two leading wearable modalities of egocentric video and inertial data, using in particular a high variety of fitness exercises and outdoor scenes. With the current selection of participants, the WEAR dataset is biased towards young, healthy people. Given the ease of reproducibility,

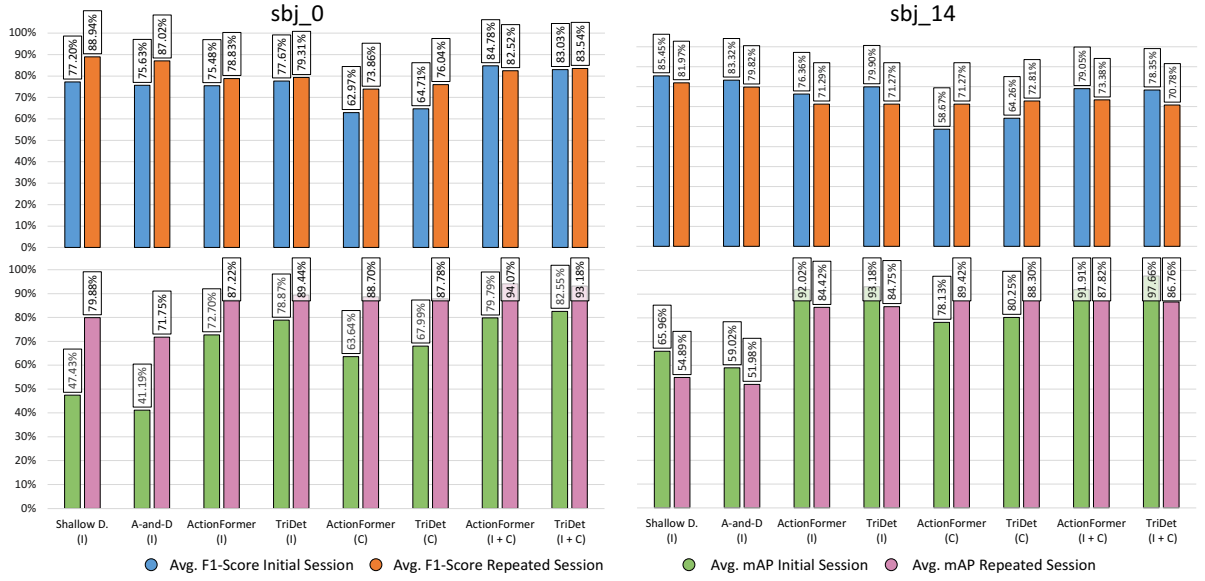


Figure 2.8.: Comparison of obtained results of repeated sessions for participants sbj_0 and sbj_14 for a clip length of 1 second on our WEAR dataset evaluated in terms of F1-score and mAP. The two participants were invited to perform the recording plan a second time. While one can see that improved results regarding sbj_0, suggesting potential learning effects of the correct execution of activities, this trend does not apply to sbj_14. Note that weather conditions (temperature and sunlight) significantly differ amongst the recordings – winter (first recording) compared to summer (2nd recording). These figures are, as in the earlier results, averaged across 3 runs using 3 different random seeds. Unlike our prior experiments, each algorithm is trained using the training data of all but the validation participants’ recordings, ensuring the validation participants (sbj_0 and sbj_14) remain unseen during the training of each algorithm. All results are postprocessed as reported in the main paper.

future extensions of the WEAR dataset could focus on featuring participants (1) of an older and/ or younger age, (2) with known physical impairments and (3) sessions recorded at new locations (outside of Germany) and at different times of the year (e.g. summer). As supplementary experiments already indicate, recording the same participants a second time would allow to analyse how a certain degree of familiarity with the recording setup can be seen in altered movements (e.g., via a smoother execution of activities) as well as give an intuition about robustness of learned approaches. Besides extending the amount of data recorded, further recordings could also involve other sensors such as higher-end commercial smartwatches to enable the study of increased sampling rates, the variability of the capturing devices, and the inclusion of additional modalities such as 3D gyroscopes, 3D magnetometers, or photoplethysmography (PPG) to obtain fitness-relevant information such as heart rate), as well as additional wearables, such as earables. Furthermore, as participants remain at the same location during their workout, including recordings of the same participants at different locations would allow to assess classifiers based on their ability to learn general cues to predict the workout activities rather than overfitting on location-specific ones. Lastly, as a camera observing the participant is needed for annotation of the collected data, a future extension of the WEAR dataset could include releasing the 3rd-person video data of the workouts along with the egocentric data, enabling the dataset to be also used e.g. for pose estimation.

2.7. Discussion & Conclusion

In this paper, we introduced a benchmark dataset for both inertial- and vision-based HAR, to explore the learning of HAR across these modalities. The dataset comprises data from 22 participants performing each 18 different sports activities with the two common types of wearable sensors delivering inertial (3D acceleration) and camera (egocentric video) data. Unlike previous egocentric datasets, the application-driven sensor setup of WEAR provides a challenging prediction scenario across both modalities marked by changing outdoor environments along with a small information overlap between the inertial and vision data, putting forward the necessity of exploring techniques to combine both modalities.

Benchmark results obtained using each modality separately show that each modality interestingly offers complementary strengths and weaknesses in their prediction performance. In light of the recent successes of single-stage TAL models following the architecture design as proposed by Zhang et al. [204], we demonstrate their versatility by applying them in a plain fashion without any pretraining using preextracted I3D features and raw inertial data as input. Using the inherent similarities of approaches of both communities, we suggest a simple, yet effective preprocessing technique of inertial data, which vectorizes sliding window such that they can be used to train TAL models. Vision-based TAL such as the

ActionFormer [204] have thus far neither been explored in inertial nor in the combination of inertial- and vision-based human activity recognition. Results show that the vision-based models are not only able to produce competitive results using inertial data, but also can function as an architecture to fuse both modalities by means of simple concatenation with vision data. In experiments that combined raw inertial with extracted vision-based feature embeddings, the plain, vision-based TAL models were able to produce the highest average mAP and close-to-best F1-scores. Lastly, to give an intuition about a possible upper bound for future fusion-approaches, we evaluated an oracle-merged late fusion of the best inertial- and vision-based model predictions.

With WEAR, we provide both communities (inertial- and vision-based HAR) a common, challenging benchmark dataset to assess the applicability of combined human activity recognition approaches.

3. Weak-Annotation of HAR Datasets using Vision Foundation Models

Title: Weak-Annotation of HAR Datasets using Vision Foundation Models

Authors: Marius Bock, Kristof Van Laerhoven, Michael Moeller

Venue: Proceedings of the 2024 ACM International Symposium on Wearable Computers (ISWC'24)

Publication date: 05 October 2024

DOI: 10.1145/3675095.3676613

Code: github.com/mariusbock/weak_har

Building on the WEAR dataset (see Chapter 2) which demonstrated the value of combining video and inertial data for activity recognition, this chapter presents a novel approach to overcome challenges associated with inertial sensor data annotation using time-synchronized video data. With data annotation remaining a tedious and time-consuming task in HAR, the work proposes a clustering-based weak-annotation pipeline that leverages discriminative feature extraction of pretrained vision foundation models. A quantitative evaluation demonstrates that resulting weakly-annotated datasets achieve labeling accuracies close to 90%, that enable the training of effective deep learning classifiers having annotated less than 10% of the original dataset. Note that the article refers to a preprint version of the WEAR dataset [19], as the dataset was under review at the time of submission.

Contextualization

As discussed in Chapter 1, the lack of interpretability in hindsight makes wearable sensor data annotation a notoriously laborious task, taking up to 14 to 20 times longer than the actual recorded data [142]. This *labeling bottleneck* of acquiring sufficient amounts of annotated data, has caused benchmark datasets used within the field of HAR to significantly lack in richness and size compared to that of related fields. With a majority of currently publicly-available inertial datasets reported to rely primarily on video recordings to allow for an accurate offline labeling process [201, 142, 149], the following work addresses the labeling bottleneck by proposing an annotation pipeline which leverages a time-synchronized video stream of the HAR experiments and vision foundation models such as CLIP [135] to significantly reduce manual labelling efforts for wearable datasets. Having previously recorded

a dataset which encapsulates a sensor-layout and recording scenario which is in line with best-practices from both the inertial and video activity recognition community, the WEAR dataset becomes a central-piece in the analysis of this work as the other two datasets tested provide only object-centric prediction tasks, recorded in a single controlled lab-environment.

While benchmarking the WEAR dataset, similarities between recognition approaches of the TAL and inertial activity recognition community were established. The following work makes use of these similarities, reapplying the TAL community’s use of pre-trained foundation models to extract latent video representations of pre-segmented clips of an input video (see Chapter 5 for a more in-depth explanation). While the TAL community uses these embeddings as input to their action localization models, the suggested per-participant annotation pipeline uses these similarities to create a clustering of a HAR experimental video, with each cluster targeted to represent an activity performed by the participant providing the data. The pipeline then identifies ”centroid” clips of each cluster by using the density surrounding each clip, with the clip with the highest density within a cluster judged to be most representative of it. In an effort to limit the annotation effort of HAR datasets, annotators using the pipeline are then required to label only these centroid clips, with labels propagated automatically to all clips within the respective cluster. Experiments within this work demonstrate that by using this annotation pipeline, one is able create weakly-annotated datasets which can even be used to train classifiers performing comparably to those trained using the fully-annotated, original dataset, having e.g. only annotated around 10% of the WEAR dataset (see Chapter 2).

Prior work conducted during this PhD, suggested that annotations provided novice annotators (i.e. persons, inexperienced with the field of activity recognition and wearable sensors) are comparable to that of expert annotators [76]. Together with these insights, we believe the proposed annotation tool described in this chapter can help work towards simplifying wearable dataset annotation. Paired with a wearable sensor setup like that in the WEAR dataset, participants could collect in-the-wild data and use their own synchronized videos for reliable self-annotation, reducing participant burden and improving long-term study retention.

The idea to address the labeling bottleneck using vision data was sparked by Michael Moeller and Kristof Van Laerhoven and was conceptualized in the DFG project WASEDO (grant number 506589320). The idea to focus on vision foundation models as well as clustering extracted latent representations of clips was proposed by Marius Bock. Implementation of the annotation pipeline, literature review and choice of experiments was conducted by Marius Bock. Kristof Van Laerhoven and Michael Moeller offered guidance and feedback throughout the implementation and enhanced the clarity and readability of the final publication during the writing process.

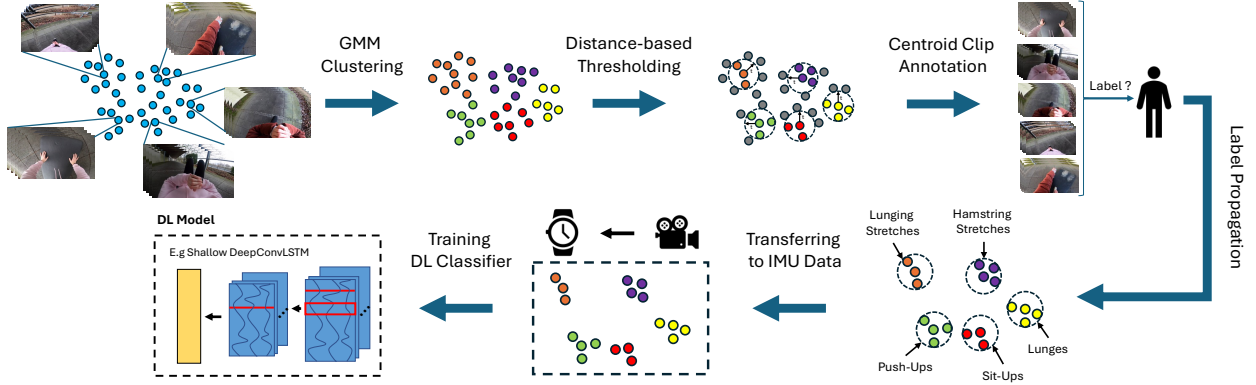


Figure 3.1.: Our proposed weak-annotation pipeline: Visual embeddings extracted using Vision Foundation Models are clustered using Gaussian Mixture Models (GMMs). Decreasing the required labelling effort, a human annotator is only asked to annotate each cluster’s centroid video clip. Centroid labels are then propagated within each cluster. Transferred to the corresponding IMU-data, resulting weakly-annotated datasets can be used to train subsequent classifiers.

Abstract

As wearable-based data annotation remains, to date, a tedious, time-consuming task requiring researchers to dedicate substantial time, benchmark datasets within the field of Human Activity Recognition in lack richness and size compared to datasets available within related fields. Recently, vision foundation models such as CLIP have gained significant attention, helping the vision community advance in finding robust, generalizable feature representations. With the majority of researchers within the wearable community relying on vision modalities to overcome the limited expressiveness of wearable data and accurately label their to-be-released benchmark datasets offline, we propose a novel, clustering-based annotation pipeline to significantly reduce the amount of data that needs to be annotated by a human annotator. We show that using our approach, the annotation of centroid clips suffices to achieve average labelling accuracies close to 90% across three publicly available HAR benchmark datasets. Using the weakly annotated datasets, we further demonstrate that we can match the accuracy scores of fully-supervised deep learning classifiers across all three benchmark datasets. Code as well as supplementary figures and results are publicly downloadable via github.com/mariusbock/weak_har.

3.1. Introduction

Though the automatic recognition of activities through wearable data has been identified as valuable information for numerous research fields [26], currently available wearable activity recognition benchmark datasets lack richness and size compared to datasets available

within related fields. Compared with, for example, the newly released Ego4D dataset [59], it becomes apparent that currently used datasets within the inertial-based HAR community are significantly smaller in terms of the number of participants, length of recordings, and variety of performed activities. One of the main drivers for this is that, even though body-worn sensing approaches allow recording large amounts of data with only minimal impact on users in various situations in daily life, wearable-based data annotation remains, to date, a tedious, time-consuming task and requires researchers to dedicate substantial time to it during data collection (taking up to 14 to 20 times longer than the actual recorded data [142]).

Following the success in other vision-related fields, researchers within the video activity recognition community have made use of feature extraction methods which provide latent representations of video clips rather than using raw image data [151, 111]. Such feature extraction methods are usually pretrained on existing large benchmark corpora, though often not particularly related, they are capable of transferring knowledge to the activity recognition task at hand. Recently, vision foundation models [135, 125] have gained a lot of attention. Typically trained on a large amount of curated and uncurated benchmark datasets, these models have helped the community further advance in finding robust, generalizable visual feature representations.

With the majority of researchers within the wearable activity recognition community relying on the vision modality to overcome the lack expressiveness of wearable data and accurately label their to-be-released benchmark datasets offline (see e.g. [28, 142, 77]), we propose a novel annotation pipeline which makes use of visual embeddings extracted using pretrained foundation models to significantly limit the amount of data which needs to be annotated by a human annotator. Our contributions are three-fold:

1. We find that visual embeddings extracted using publicly-available vision foundation models can be clustered activity-wise.
2. We show that annotating only one clip per cluster suffices to achieve average labelling accuracies above 60% and close to 90% across three publicly available HAR benchmark datasets.
3. We demonstrate that using the weakly annotated datasets, one is capable of matching accuracy scores of fully-supervised deep learning classifiers across all three benchmark datasets.

3.2. Related Work

Vision Foundation Models The term foundation models was coined by Devlin et al. [44] and refers to models which are pre-trained on a large selection of datasets. The idea of

pre-training models on large benchmark datasets has been prominent within the vision community for a long time. Within the video classification community researchers demonstrated that pretrained methods such as I3D [27], VideoSwin [113] or SlowFast [51] extract discriminate feature embeddings which can be used to train subsequent classifiers. Following their success in Natural Language Processing [25, 44], researchers applied masked autoencoders on visual data input. Unlike previous methods, masked autoencoders are capable of pretraining themselves in a self-supervised manner, allowing the use of larger data sources. Two of such methods are CLIP [135] and DINOv2 [125]. The former, published by OpenAI, is a vision-language model which tries to learn the alignment between text and images. According to the authors, CLIP is pretrained on a large corpora of image-text pairs scraped from the world wide web. Similarly, the recently released DINOv2 by META AI makes an effort of providing a foundation model which is capable of extracting general purpose visual features, for which the authors collected data from curated and uncurated sources.

Weakly-Supervised Wearable HAR With the activity labelling of body-worn sensor data being a tedious task, many researchers have looked at weakly-supervised learning techniques to reduce the required amount of annotations to train subsequent classifiers. Early works such that of Stikic et al. [157] have shown to reduce the labelling efforts for training classical machine learning models through knowledge-driven approaches using graph-based label propagation [158], multi-instance learning [159] or probabilistic methods [55, 168]. Adaimi and Thomaz [3] followed the works of [157], proposing an active learning framework which focuses on asking users to label only data which will gain the most classification performance boost. With the rise in popularity of Deep Learning, deep clustering algorithms have been proposed to cluster latent representations in unsupervised and semi-supervised fashion using e.g. autoencoders [101, 179, 5, 117, 7], recurrent networks [55, 2], self-supervised [145] and contrastive learning [4, 190, 171, 41, 91]. Recently, Xia et al. [190] and Tong et al. [171] demonstrated how vision-foundation models such as CLIP [135] and I3D [27] can be used to create visual, complementary embeddings to inertial data such that a contrastive loss can be calculated. This work marks one of the few instances of researchers trying to use visual data to limit the amount of annotations required in wearable activity recognition. Our work ties into the works of Tong et al. [171] and Xia et al. [190], yet we propose instead to apply vision foundation models to perform automatic label propagation between similar embeddings.

Table 3.1.: Network layer used for extracting embeddings of the different vision foundation models [27, 135, 125]. Subsequent layers are omitted such that the network outputs latent representations at point of the embedding layer. Note that the I3D network is used for extracting both RGB and flow features and we refer to the vision-based part of the CLIP model.

Model	Embedding Layer	Dimension
I3D	last average pool layer	$\mathbb{R}^{1024}$
CLIP	last projection layer (vision-CLIP)	$\mathbb{R}^{768}$
DINO	last layer hidden state (clf. token)	$\mathbb{R}^{1024}$

3.3. Methodology

3.3.1. Annotation Pipeline

Latent Space Clustering via Vision Foundation Models Within the first phase we divide the unlabeled dataset into (overlapping) video clips. Given an input video stream X of a sample participant, we apply a sliding window approach which shifts over X , dividing the input data into video clips, e.g. of four second duration with a 75% overlap between consecutive windows. This process results in $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ being discretized into $t = \{0, 1, \dots, T\}$ time steps, where T is the number of windows, i.e. video clips, for each participant. Inspired by classification approaches originating from the temporal action localization community, we make use of pretrained vision foundation models to extract latent representations of each clip. That is, $\mathbf{x}_t \in \mathbb{R}^E$ represents a one-dimensional feature embedding vector associated with the video clip at time step t , where E is number of latent features the embedding vector consists of. In total we evaluated three popular pretrained foundation models: a two-stream inflated 3D-ConvNet (I3D) [27] pretrained on the RGB and optical flow features extracted from Kinetics-400 dataset [90] as well as two transformer foundation models CLIP [135] and DINOv2 [125], which were pretrained on a multitude of curated and uncured data sources. Note that, unlike Carreira and Zisserman in [27], we use RAFT [169] instead of TV-L1 [147] optical flow estimation. As the CLIP and DINOv2 model both are not explicitly trained on optical flow features, we also test complementing embeddings of the two models by concatenating them with extracted embeddings of the inflated 3D-ConvNet trained on RAFT optical flow features of the Kinetics dataset. In order to obtain latent representations we altered models such that intermediate feature representations can be extracted. Table 3.1 provides details activations at which layer were considered to be the embedding of which pretrained method as well as their dimension. To merge together the frame-wise features outputted of the CLIP and DINOv2 model, we apply average pooling as detailed in [116] to obtain a single latent representation per sliding video clip.

Having extracted latent representations of each video clip within the tested benchmark datasets, we apply GMM to cluster the embeddings on a per-participant level. Though GMMs are not originally intended to be used with high-dimensional data, they have shown to provide good results clustering visual embeddings especially in the context of action recognition [95, 178] and allow, unlike methods such as k-nearest neighbors, more flexibility regarding the shape of clusters. Training one GMM clustering algorithm per participant and applying it to said data assigns each video clip $\mathbf{x}_t \in \mathbb{R}^E$ a cluster label $x_c \in 1, \dots, C$, where C is the number of GMM components, i.e. clusters, applied.

Weak-Labeling via Centroid Clips Once each video clip of a study participant has been assigned a cluster label x_c , the second phase of our approach consists of a human annotator only needing to annotate one sample clip per cluster. Assuming the centroid of a cluster is most representative of all clips within that cluster, we can propagate the activity label $a \in 1, \dots, A$ of said clip to all other clips eliminating the need of annotating the other clips via a human annotator, where A is the number of activities within the dataset. As GMM do not explicitly provide a definition of a centroid of a component, we calculate the centroid clip of each cluster component being the clip which has the highest density within said cluster. That is, given the covariance matrix of each mixture component $\Sigma \in \mathbb{R}^{E \times E}$, assuming each component has its own general covariance matrix, and mean vector $\mu \in \mathbb{R}^E$, we calculate the density of each point as the logarithm of the probability density function of the multivariate normal distribution defined by μ and Σ . Having identified the centroid clip within each cluster, our approach propagates the annotation provided by the human annotator to all other clips, which were also assigned to that cluster.

As our approach forces each video clip to be assigned an activity label, we augment our clustering with a subsequent distance-based thresholding in order to remove outlier clips from the automatic labelling. Assuming that the distance of another clip to the centroid resembles its likelihood of belonging to the same activity class, we omit clips from the dataset which exceed a certain distance from their respective centroid clip, with the distance being calculated as the L^2 -norm between two embedding vectors. Even though this approach decreases the amount of data which can be used to train subsequent classification algorithms, we show to increase the overall labelling accuracy by a significant margin.

3.3.2. Weakly-Supervised Training

Assuming inertial and video data are synchronised, we further evaluate how well the resulting annotated inertial data with non-uniform label noise is suited to be used for training inertial-based deep learning classifiers. As our benchmark algorithms of choice we use two

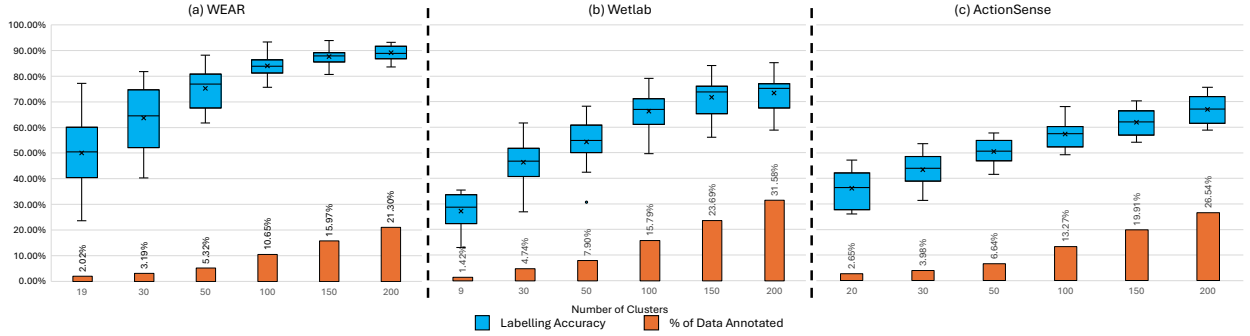


Figure 3.2.: Box-plot diagrams ● showing the distribution of labelling accuracies across study participants with increasing number of clusters. The bar plot ● below the box-plots provides details per cluster setting about the percentage of data compared to the total size of the three benchmark datasets [19, 149, 42] an annotater would need to annotate. One can see a clear trend that with an increase in clusters, labelling accuracy increases along with deviation across study participants decreasing.

recently published state-of-the-art methods, namely the Shallow DeepConvLSTM [17] and TinyHAR architecture [210]. We use both architectures as originally introduced by the authors, specifically using the same size and number of convolutional filters, convolutional layers and size of the recurrent layers. During training we apply a sliding window of one second with an overlap of 50%, as it proved to be provide consistent classification performances across a multitude of HAR datasets [17]. We train each network for 30 epochs using the Adam optimizer (learning rate $1e^{-4}$ and weight decay $1e^{-6}$) applying a step-wise learning rate with a decay factor of 0.9 after each 10 epochs. To mitigate the introduced label noise by our proposed weak-annotation pipeline, we calculate the loss during training using the weighted partially Huberised generalised cross-entropy (PHGCE) loss [118], which extends the definition of the generalized cross-entropy loss [206] with a variant of gradient clipping. To compare the validity of our approach, we compare amongst a set of (weakly-)annotated training approaches:

1. *Fully-supervised*: Fully-supervised results using the original, fully-annotated benchmark datasets.
2. *Few-Shot-CE*: Fully-supervised training using only the annotated clips and a weighted cross-entropy loss.
3. *Random-CE*: training using an equal amount of random annotated clips as in (2) and a weighted cross-entropy loss.
4. *Weak-CE*: Weakly-supervised training using the weakly-annotated dataset and a weighted cross-entropy loss.
5. *Weak-PHGCE*: Weakly-supervised training using the weakly-annotated dataset and a weighted PHGCE loss.

3.3.3. Datasets

WEAR The WEAR dataset offers both inertial and egocentric video data of 18 participants doing a variety of 18 sports-related activities, including different styles of running, stretching, and strength-based exercises. Recordings took place at changing outdoor locations. Each study participant was equipped with a head-mounted camera and four smartwatches, one worn on each limb in a fixed orientation, which captured 3D-accelerometer data.

ActionSense Published by DelPreto et al. [42], the ActionSense dataset provides a multitude of sensors capturing data within an indoor, artificial kitchen setup. Amongst the sensors, participants wore IMUs on both wrists as well as smart glasses which captured the ego-view of each participant. During recordings, participants were tasked to perform various kitchen chores including chopping foods, setting a table and (un)loading a dishwasher. Within their original publication, the authors provide annotations of 19 activities of 10 participants. Note that the dataset download of the ActionSense dataset provides IMU and egocentric video data of only 9 instead of 10 participants.

Wetlab Taking place in a wetlab laboratory environment, the Wetlab dataset [149] comprises of data of 22 study participants which performed two DNA extraction experiments. For purposes of this paper we used the annotated provided by the authors of the reoccurring activities base activities (such as stirring, cutting, etc.) within the experimental protocol. During recordings, each participant wore a smartwatch in a fixed orientation on the wrist of their dominant hand, which captured 3D-accelerometer data. Unlike the WEAR and ActionSense dataset, the Wetlab dataset provides video data of a static camera which was mounted above the table at which the experiment was performed, thus capturing a birds-eye perspective of the experiment’s surroundings.

3.4. Results

To ensure that reported performance differences are not the based on statistical variance, all reported experiments are repeated three times, applying a set of three predefined random seeds. This applies both for the annotation pipeline experiments as well as weakly-supervised training results. During all annotation-based experiment mentioned in Section 3.4.1 we apply a clip length of four seconds along with a three second overlap between clips. We assume that this clip length is a suitable length to be interpretable for a human annotator while simultaneously avoiding mixing multiple activities into one sliding window. Furthermore, during all experiments only the label of the centroid clip is propagated to that of all other cluster instances. Ablation experiments evaluating different clip lengths and number of

Table 3.2.: Average labeling accuracy and standard deviation across study participants using different types and combinations of embeddings [27, 135, 125] extracted from three benchmark datasets [19, 149, 42] applying GMM-based clustering using 100 clusters. Overall, a combination of CLIP and optical flow embeddings proved most consistent across all datasets.

	WEAR	Wetlab	ActionSense
(1) I3D	82.62 (± 4.65)	66.08 (± 9.53)	53.47 (± 5.95)
(2) CLIP	82.47 (± 6.03)	72.70 (± 6.42)	59.85 (± 4.42)
(3) DINOv2	79.20 (± 4.04)	69.28 (± 8.12)	60.25 (± 4.04)
(4) RAFT	76.86 (± 4.79)	51.50 (± 6.96)	45.64 (± 5.19)
(1) + (4)	85.17 (± 4.48)	60.91 (± 8.36)	53.00 (± 4.66)
(2) + (4)	83.96 (± 4.99)	66.23 (± 7.86)	57.29 (± 5.64)
(3) + (4)	79.13 (± 4.30)	70.18 (± 9.79)	56.55 (± 4.51)

annotated clips per cluster to determine the label to be propagated can be found within our code repository.

3.4.1. Annotation Pipeline

Table 3.2 shows the average labelling accuracy averaged across participants obtained when applying our proposed annotation pipeline using various types of extracted visual embeddings. One can see that in case of the WEAR [19] and ActionSense dataset [42] labelling accuracy can be improved by combining both RGB and optical flow features in case of all embeddings. Overall a combination of CLIP and optical flow features proves to be most consistent across our three benchmark datasets of choice, making it thus our embedding of choice for subsequent experiments. Applying a labelling strategy of only annotating the centroid clip of each cluster, Figure 3.2 presents a box-plot visualization of applying different number of clusters during the clustering of the participant-wise embeddings. One can see that by only annotating 100 clips per study participant, our proposed annotation pipeline is capable of reaching labelling accuracies above 85% in case of the WEAR and close to 70% in case of the Wetlab [149] and ActionSense dataset.

Furthermore, as evident by an overall shrinking boxplot with increasing number of clusters, our approach is becoming more stable with the standard deviation across study participants decreasing in case of all three datasets. As Figure 3.2 shows, applying a clustering of $C = A$, i.e. as many clusters as there are activities in the dataset, results in the clustering not being capable of differentiating the normal and complex variations of activities, different running styles and null-class from all other activities. In general, we witness a trend that by applying a larger amount of clusters than activities present in the dataset, one gives the GMM clustering enough degrees of freedom to differentiate even activities which share

similarities, yet slightly differ from each other. Lastly, by distance thresholding clusters and excluding instances which are exceeding a certain distance from their respective centroid, helps increasing the labelling accuracy significantly across all datasets. While a threshold of 4 helps increase the labelling accuracy well above 75% and even up to 93% in case of the WEAR dataset, the thresholding omits between 50% and up to 90% of the datasets.

3.4.2. Weakly-Supervised Training

As a combination of CLIP and optical-flow-based features proved to be most stable across all three datasets, we chose to use said embedding as basis for our weakly-supervised training. Table 3.3 provides an overview across the eight evaluated training scenarios. Our proposed weakly-supervised training is not only capable of outperforming the few-shot training using only the annotated centroid clips, but for the case of applying 100 clusters is close to matching accuracy scores of a fully-supervised training across all three benchmark datasets, for both inertial-based architectures. Compared to a normal cross-entropy loss, the PHGCE loss provides more stable results in case of higher label noise, e.g., when not applying a distance-based thresholding and/or applying a smaller number of clusters. In general, the distance-based thresholding significantly improved results across all datasets. Although thresholding significantly reduces the amount of training data, the resulting decrease in overall labelling noise, especially for approaches that applied a lower number of clusters, improved classification results. We provide a detailed overview of the influence of thresholding on labelling accuracy and dataset size within the paper’s code repository.

Figure 3.3 shows a side-by-side comparison of the confusion matrices of the fully-supervised and best-performing weak-labelling approach (Weak-CE-T-4) using the WEAR dataset and shallow DeepConvLSTM. One can see that, apart from the NULL-class, classification results of all activities of the weakly-supervised training are similar to that of the fully-supervised. With the intra-class similarity of a NULL-class within datasets being quite low, it is to a larger degree grouped together with instances of other classes making it harder for an inertial-based classifier to learn unique patterns only applicable to that of the NULL-class. Looking at per-class results of the Wetlab dataset, one can see that the introduced labelling noise caused the classifier trained using weakly-annotated data is more likely to confuse activities with NULL-class rather than vice versa. This caused the activities *transfer* and *pouring*, two classes which only have a few instances in the ground truth data and which are most frequently annotated incorrectly as NULL-class, to not be predicted correctly once across all study participants. Note that for the ActionSense dataset classification results even in the fully-supervised setting are significantly worse compared to that of the other two datasets. As evident by a nevertheless large labelling accuracy using our approach, we assume that

label semantics of the dataset are too vision-centric (e.g. peeling a cucumber or a potato) to be correctly recognized using only inertial data. Nevertheless, per-class classification results of the fully- vs. weakly-supervised training show a similar confusion amongst classes, suggesting learned patterns of the classifier are similar in both training scenarios.

3.5. Discussion & Conclusion

Within this paper we presented a weak-annotation pipeline for HAR datasets based on Vision Foundation Models. We showed that visual embeddings extracted using Vision Foundation Models can be clustered using GMM. Decreasing the required labelling effort, using the suggested pipeline a human annotator is only asked to annotate each cluster’s centroid video clip. By propagating the provided labels within each cluster our approach is capable of achieving average labelling accuracies above 60% and close to 90% across three popular HAR benchmark datasets. We further showed that the resulting weakly-annotated wearable datasets can be used to train subsequent deep learning classifiers with accuracy scores, in case of applying a sufficiently large number of clusters, being close to matching that of a fully-supervised training across all three benchmark datasets.

Our results underscore one of the implications recent advancements in the vision community in finding generalizable feature representations might have on the field of HAR. With the rapid progress being made in the area of foundation models, follow-ups of models such as CLIP and DINOv2 could further robustify the automatic analysis of collected video streams in wearable-based data collection. Our clustering-based pipeline thus not only improves the efficiency of data annotation but also contributes to the creation of richer and more extensive HAR benchmark datasets.

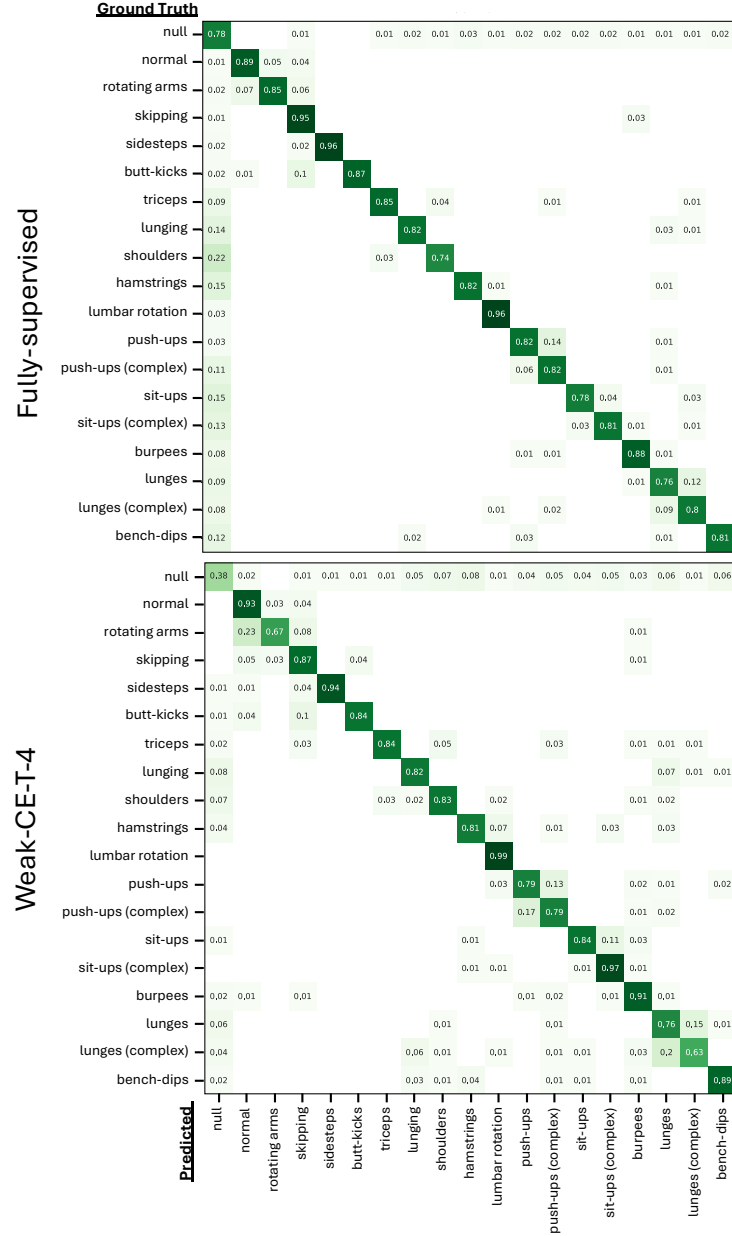


Figure 3.3.: Confusion matrices comparing the shallow DeepConvLSTM fully-supervised results compared to that of the best performing weak-labelling approach. With exception of the NULL-class, all activities were able to be classified close to the performance of the fully-supervised approach.

4. Improving Deep Learning for HAR with shallow LSTMs

Title: Improving Deep Learning for HAR with shallow LSTMs

Authors: Marius Bock, Alexander Hoelzemann, Michael Moeller, Kristof Van Laerhoven

Venue: Proceedings of the 2021 ACM International Symposium on Wearable Computers (ISWC’21)

Publication date: 21 September 2021

DOI: 10.1145/3460421.3480419

Code: github.com/mariusbock/dl-for-har

This work marked the first peer-reviewed contribution of the dissertation project. It revisits the widely-used DeepConvLSTM model by Ordóñez and Roggen [126] and proposes a lighter and more performant alternative: the Shallow DeepConvLSTM. Contrary to the popular belief temporal learning within sequential data requires at least two LSTM-layers [89] and the overall trend towards more complex temporal modeling techniques, this work demonstrates that employing a single LSTM-layer in the DeepConvLSTM, as opposed to two layers, yields significantly better performance across various HAR datasets, with improvements up to 11.7% in F1-score. This work received the Best Note Paper Award at ISWC’21.

Contextualization

The following work originated within the initial efforts of developing a modernized version of the Activity Recognition Chain (ARC), originally introduced by Bulling et al. [26] (see Chapter 1 for further contextualization). The updated pipeline, referred to as the Deep Learning Activity Recognition Chain (DL-ARC), was presented in a tutorial at ISWC’21 by the same authors as the following work [15], aiming to reflect the current state-of-the-art in applying Deep Learning to inertial-based activity recognition. The proposed DL-ARC centers around the DeepConvLSTM [126] as the deep learning architecture of choice. The architecture, originally introduced by Ordóñez and Roggen in 2016, combined convolutional and recurrent layers to both extract and temporally model discriminative features from raw inertial data. Being frequently used during benchmarks for inertial-based human activity recognition, the architecture remains to date a viable choice. While research following the

initial release of the DeepConvLSTM has sought to enhance models’ capabilities to better model temporal patterns through the integration of mechanisms like attention [210], the sliding window problem fundamentally constrains these models in temporal understanding to within-window patterns [26].

Opposing the trend of utilizing more computationally expensive temporal modeling techniques, early results using the Wetlab dataset [149] and the DL-ARC revealed significant performance improvements when reducing the number of LSTM layers within the DeepConvLSTM from two to one. This finding further contradicts prior claims such as those by Chen et al. [31] referencing Karpathy et al. [89] that multiple LSTM layers are essential for effective temporal modeling when dealing with sequential data. The following work thus presents a systematic study comparing 1-layer and 2-layer LSTM variants within the original DeepConvLSTM. The resulting Shallow DeepConvLSTM, employing a 1-layered LSTM, is marked by a significant decrease in number of learnable parameters and training time, while consistently outperforming the original DeepConvLSTM across five benchmark datasets. The publication surrounding the work received the Best Note Paper Award at ISWC’21.

Further inspection of the subsequent work after publishing revealed an unexplored change: the LSTM was applied across the sequence of windows within a batch, rather than across time steps within each sliding window. This led the investigated models to learn inter-window rather than intra-window temporal dependencies, deviating from conventional batch semantics. Despite this, the model still achieved strong recognition performance, suggesting the potential value of inter-window learning as it enables the model to learn from longer temporal sequences, spanning multiple windows. Inspired by these findings as well as results discussed in Chapter 5, the following paper was revisited in a follow-up work, described in Chapter 6, which was under review at the time of this dissertation submission. The paper demonstrates that initial findings about reducing the depth of LSTMs, which will subsequently be discussed, still hold for both intra- and inter-window temporal learning and suggests a novel architecture, called DeepConvContext, which combines both types of temporal learning.

The idea to work on an updated ARC was suggested by Kristof Van Laerhoven. Upon presenting benchmark results, Marius Bock noticed the increased performance by shallow LSTMs. Subsequent experiments and literature review were conducted by Marius Bock. Alexander Hoelzemann, Kristof Van Laerhoven and Michael Moeller provided guidance and feedback throughout the implementation. Kristof Van Laerhoven and Michael Moeller enhanced the clarity and readability of the final publication during the writing process. This work was further refined through an anonymous shepherding process as part of publishing at the *ISWC’24* conference suggesting the inclusion of the Opportunity dataset [142] in the benchmark analysis.

Abstract

Recent studies in HAR have shown that Deep Learning methods are able to outperform classical Machine Learning algorithms. One popular Deep Learning architecture in HAR is the DeepConvLSTM. In this paper we propose to alter the DeepConvLSTM architecture to employ a 1-layered instead of a 2-layered LSTM. We validate our architecture change on 5 publicly available HAR datasets by comparing the predictive performance with and without the change employing varying hidden units within the LSTM layer(s). Results show that across all datasets, our architecture consistently improves on the original one: Recognition performance increases up to 11.7% for the F1-score, and our architecture significantly decreases the amount of learnable parameters. This improvement over DeepConvLSTM decreases training time by as much as 48%. Our results stand in contrast to the belief that one needs at least a 2-layered LSTM when dealing with sequential data. Based on our results we argue that said claim might not be applicable to sensor-based HAR.

This paper’s architecture and experiment scripts are publicly downloadable via <https://github.com/mariusbock/dl-for-har>.

4.1. Introduction

Physical activities play a crucial role in the way we structure our lives. Which activity, and how it is performed, can reveal a person’s intention, habit, fitness, and state of mind; it is therefore not surprising that a range of research fields, from cognitive science to healthcare, display a growing interest in the machine recognition of human activities, also known as HAR [26]. Deep Learning methods have in the past decade shown to outperform classical Machine Learning algorithms (e.g., [94, 37, 163]) and, as a product of this success, have led to studies investigating the effectiveness of Deep Learning in HAR (e.g., [126, 65]).

One of the most popular Deep Learning architectures for HAR is the DeepConvLSTM which was proposed by Ordóñez et al. in [126]. The suggested architecture combines both recurrent and convolutional layers and received state-of-the-art results on both the Opportunity [142] and the Skoda Mini Checkpoint [201] dataset. The original DeepConvLSTM architecture employs a 2-layered LSTM with 128 hidden units. A common belief, as e.g. stated by Chen et al. [31], based on the findings of Karpathy et al. [89], is that one requires at least a 2-layered LSTM when dealing with sequential data. With this paper, we aim at challenging this belief and suggest that re-examining the architecture of the DeepConvLSTM by employing a one-layered LSTM, has considerable benefits.

Our paper’s contributions are threefold:

1. We show that an altered DeepConvLSTM architecture with a one-layered LSTM overall

outperforms architectures employing a two-layered LSTM, by validating our claim using the Opportunity dataset [142] as seen in [126], as well as 4 other popular HAR datasets [149, 164, 140, 160].

2. Using our suggested architecture change, we reduce the number of learnable parameters within the DeepConvLSTM and thus are able to decrease training time significantly.
3. We provide our PyTorch-based architecture, experiment scripts and log files in a GitHub repository for others to replicate these findings and continue such analyses.

4.2. Related Work

Deep Learning in HAR. The predictive performance of classical Machine Learning approaches highly relies on sophisticated, handcrafted features [133]. In the last decade, Deep Learning has shown to outperform classical Machine Learning algorithms in many areas, e.g. image recognition [50, 170, 163], speech recognition [94, 120, 72, 146] and Natural Language Processing [37, 24, 86, 162]. Much of this success can be accredited to the fact that Deep Learning does not require manual feature engineering, but is able to automatically extract discriminative features from raw data input [123]. The advantage of being able to apply algorithms on raw data and not being dependent on handcrafted features has led to studies investigating the effectiveness of Deep Learning in HAR, which e.g. suggested different architectures [126, 64, 63, 194, 191, 35, 196, 143, 99, 62], evaluated the generality of architectures [65] and assessed the applicability in real-world scenarios [98, 64, 87, 61].

DeepConvLSTM. One popular HAR Deep Learning architecture is the DeepConvLSTM which combines both convolutional and Long-Short-Term-Memory (LSTM) layers [126]. By combining both types of layers the network is able to automatically extract discriminative features and model temporal dependencies. The idea of Ordóñez et al. [126] to combine recurrent and convolutional layers served as the basis for several further improvements in subsequent works. For example, Murahari and Plötz [121] improved results by appending attention layers to the original architecture. Xi et al. [188] improved the performance of the original DeepConvLSTM architecture by adding dilated convolution layers in addition to normal convolution layers. Demonstrating a different way of combining recurrent and convolutional layers, Xu et al. [191] proposed *InnoHAR* which combines Inception modules based on GoogLeNet [163] and Gated Recurrent Unit (GRU) layers. Xia et al. [189] suggested to first apply a 2-layered LSTM whose output is then fed to two convolution layers and extended the architecture with a max pooling, global average pooling and batch normalization layer. Kim and Cho [92] make use of the architecture proposed by [209] which, similar to our approach, employs a one-layered LSTM. To our knowledge the work of Kim

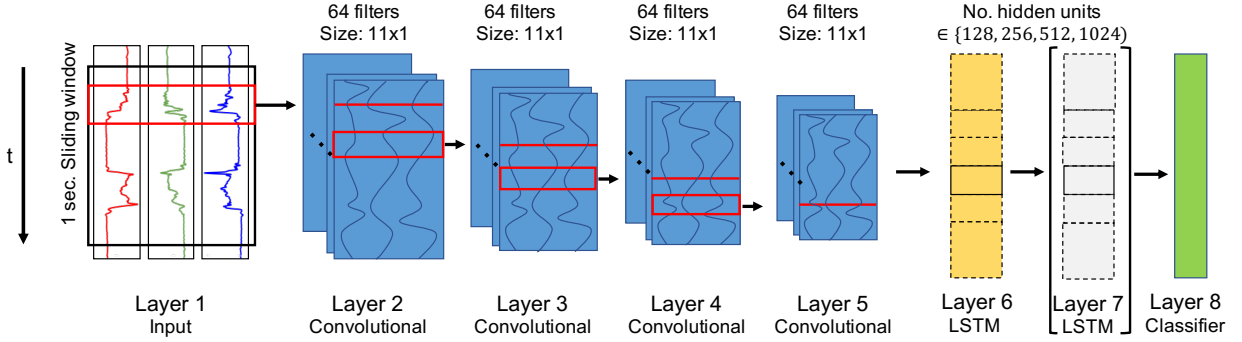


Figure 4.1.: Illustration of the suggested change to the DeepConvLSTM [126] architecture. The change involves removing the second LSTM layer (Layer 7). During experiments we either include (original architecture) or exclude Layer 7 (our suggestion) and vary the amount of hidden units within the LSTM layer(s) (i.e. Layers 6 and 7) to be either 128, 256, 512 or 1024.

and Cho [92] is the only work which applied a one-layered LSTM within a variation of the DeepConvLSTM architecture.

LSTMs and sequential data. Upon the experiments conducted by Karpathy et al. [89], Chen et al. claim within their recent survey paper that "the depth of an effective LSTM-based RNN needs to be at least two when processing sequential data" [31]. Within this paper we investigated the effect of employing a 1-layered instead of a 2-layered LSTM within the DeepConvLSTM architecture. As Karpathy et al. [89] obtained their results using character-level language models, i.e. text data, our paper aims at challenging the belief that their claim is applicable to sensor-based HAR.

4.3. Methodology

Contrary to the belief that one needs at least a two-layered LSTM when dealing with sequential data [89], we propose to change the DeepConvLSTM to have a one-layered LSTM. Figure 4.1 illustrates the suggested architecture change. We further investigate the varying amount of hidden units in the LSTM layers during experiments.

4.3.1. Datasets

To validate our architecture change, we chose to use the preprocessed version of the Opportunity dataset [142] as used by Ordóñez et al. [126], as well as four popular HAR datasets, namely the Wetlab [149], RealWorld HAR (RWHAR) [164], Smartphone-Based Recognition of Human Activities and Postural Transitions (SBHAR) [140] and the Heterogeneity Activity Recognition (HHAR) [160] dataset.

Opportunity. The Opportunity datasets consists of 4 individuals performing a set of

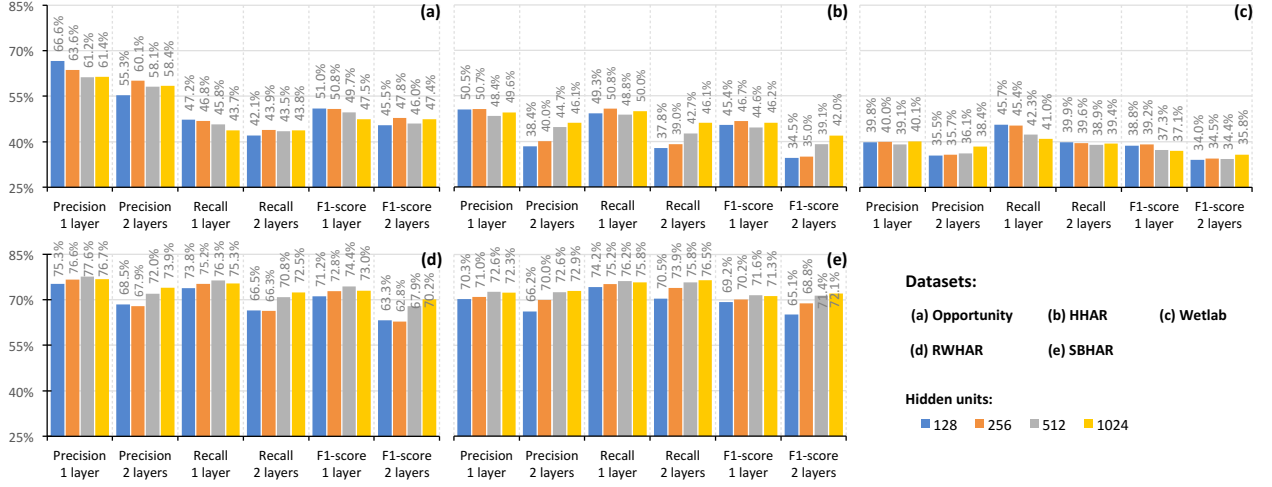


Figure 4.2.: Results obtained on the (a) Opportunity, (b) HHAR, (c) Wetlab, (d) RW HAR and (e) SB HAR datasets using the 8 variations of the DeepConvLSTM [126] architecture, grouped by 1- and 2-layered architectures and color-coded by how many hidden units were employed in the LSTM layer(s). Results are given as the average precision, recall and F1-score across 5 runs using a set of 5 varying seeds. The 1-layered architecture variants continuously outperform the 2-layered ones for the HHAR, Wetlab and RW HAR dataset; The worst performing 1-layered variant even still outperforms the best performing 2-layered variant. For the Opportunity dataset, the 1-layered variant outperforms the 2-layered for 128, 256 and 512 hidden units and is on par for 1024 hidden units. For the SB HAR dataset, the 1-layered variant only outperforms for 128 and 256 hidden units and is on par with the 2-layered version when using 512 and 1024 hidden units.

activities of daily living [142]. For the gesture recognition challenge of the dataset, there are 18 classes which are to be predicted (*open/close door 1 and 2, fridge, dishwasher and drawer 1, 2 and 3, clean table, drink from cup and toggle switch*) as well as a *null* class. During the experiments, we applied the same preprocessing as well as train-test split as suggested by Ordóñez et al. [126]. The resulting preprocessed dataset consists of in total 113 feature channels each representing an individual sensor axis from body-worn accelerometers and inertial measurement units (combining 3D accelerometers, gyroscopes and magnetometers).

Wetlab. The Wetlab dataset consists of 22 participants performing two DNA extraction experiment within a wetlab environment [149]. During experiments subjects were equipped with a wrist-worn sensing unit capturing 3D acceleration data with a sampling rate of $50Hz$. Using recorded video footage, Scholl et al. identified 9 different actions (*cutting, inverting, peeling, pestling, pipetting, pouring, stirring, transfer*) which, along the *null* class, are the target labels to be predicted within this dataset.

RealWorld (HAR). The RW HAR dataset contains data of 15 participants performing 8 different activities (*walking upstairs, walking downstairs, jumping, lying, standing, sitting, running, walking*) as well as a *null* class [164]. In order to make results more comparable to the Wetlab dataset, we chose to only use 3D acceleration data captured by a wrist-worn sensor which samples the data at $50Hz$.

SBHAR. The SBHAR dataset consists of 30 participants performing activities of daily living (*standing, sitting, lying, walking, walking downstairs, walking upstairs*) [140]. In addition to the 6 activities, class labels also include 6 postural transitions (*stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, lie-to-stand*) as well as a *null* class. For the same reason as mentioned with the RWHAR dataset, we only used the raw 3D acceleration sensor data, which was sampled at $50Hz$.

HHAR. Similar to the RWHAR dataset, the HHAR dataset contains data of 9 human participants performing activities of daily living. There are 6 activities (*biking, sitting, standing, walking, walking upstairs and downstairs*) and a *null* class which are to be predicted [160]. As for the previous datasets, we only use the 3D-acceleration data obtained from a wrist-worn sensor unit, which was sampled at $100Hz$.

4.3.2. Training

To justify our architecture and illustrate the effectiveness of said change to it, we compare two variations of the DeepConvLSTM architecture with each other: One equipped with a single-layer LSTM, and one equipped with a two-layer LSTM. We further vary the number of hidden units, more specifically using in both variations either 128, 256, 512 or 1024 hidden units per LSTM layer. This leaves us with a total of 8 variations that were evaluated, for all five datasets. It is important to note that only the LSTM was altered. Specifications of other layers, such as the number of convolution layers (4), the number of convolution filters (64) and the dropout rate (0.5), were left unchanged from the original DeepConvLSTM architecture.

To train our networks we employ a sliding window approach similar to the one as seen in [126]. To obtain suitable set of hyperparameters we evaluated multiple settings based on results obtained from using the Wetlab dataset as input. We identified a sliding window of 1 second with a 60% overlap to be most suitable. Unlike Ordóñez et al. [126] we use the *Adam* optimizer with a smaller weight decay ($1e^{-6}$) and learning rate ($1e^{-4}$) and initialize the network weights using the Glorot initialization [57]. During training we computed the loss using a weighted cross-entropy loss to enable the networks to learn also imbalanced datasets (e.g. Wetlab). Since the focus of our paper lies on evaluating the change to the architecture, we did not perform any hypertuning on the RWHAR, SBHAR and HHAR dataset and thus kept hyperparameters consistent across datasets. Nevertheless, unlike the other 3 datasets, the HHAR was sampled at $100Hz$. We thus increased the convolutional filter size by a factor of two to be 21, to maintain the relation between convolutional filter and sliding window size and capture the same amount of information with each filter across all datasets. We used the same set of hyperparameters for the Opportunity dataset as was used for the other

datasets to allow a comparison across the different types of activity recognition scenarios, and changed the sliding window size and overlap to be identical with the one employed in [126], i.e. 0.5 seconds with a 50% overlap.

4.3.3. Results

For the Wetlab, RWHAR, HHAR, SBHAR and HHAR dataset, results were obtained using Leave-One-Subject-Out (LOSO) cross-validation. This means that each subject was being used as the validation set exactly one time while all other subjects were used as training data. The final validation results are then the average across all subjects. Using LOSO cross-validation ensures that obtained results were not a product of overfitting on subject-specific traits. As we are using the same preprocessed version of the Opportunity dataset as introduced in the original work [126], we also employed the same train/ test split and trained the networks using said inputs.

Figure 4.2 shows results for all five datasets using varying hidden units and either a 1- or 2-layered LSTM within the DeepConvLSTM architecture. Results were obtained by averaging across 5 runs with a set of 5 varying random seeds. We report standard evaluation metrics, namely precision, recall and F1-score.

As we can see in Figure 4.2 architectures which employ a 1-layered LSTM overall outperforms architectures which employ a 2-layered LSTM across four (Opportunity, Wetlab, HHAR and SBHAR) out of five datasets. Only for the case of 1024 employed hidden units and using the Opportunity dataset as input, our architecture is on par with the original DeepConvLSTM architecture. For the Wetlab, RWHAR and HHAR dataset we can even see that the best performing 2-layered LSTM architecture variation performs worse, as far as precision, recall and F1-score, than the worst performing 1-layered LSTM architecture variation. Only for the SBHAR dataset our architecture is marginally worse than the original one when using 1024 hidden units within the LSTM layers.

By removing the second LSTM layer we are able to get performance increases up to 11.7% in F1-score (HHAR, 256 hidden units). Looking at Figure 4.2 one can see that the performance increase is the largest for variations employing 128 and 256 units and steadily decreases for larger amounts of hidden units. While one can see that the performance difference was not as significant for the SBHAR dataset when employing 512 hidden units and even slightly negative for the 1024 hidden units architecture, we nevertheless argue that both architecture variations are at least on par due to the fact that one has to consider statistical variance. Over the five runs, we witness an average standard deviation of 1.67% (1.53%) for precision, 1.69% (1.44%) for recall and 1.65% (1.38%) for the F1-score, using the 1-layered (2-layered) version of the architecture, respectively. The exact averages per architecture vari-

ation can be found in the repository (see <https://github.com/mariusbock/dl-for-har>).

As HAR datasets are relatively small in size compared to other popular Deep Learning datasets, e.g. [43], larger networks are more likely to be prone to overfitting. Though we witnessed this trend for the HHAR and Opportunity dataset, by monitoring the generalization gap (i.e. the difference in loss, accuracy, precision, recall and F1-score between the train and validation set), we did not see any indication that the 2-layered architecture variations are more likely to overfit and saw both architectures equally suffering from this phenomenon.

With the suggested removal of one LSTM layer we are decreasing the complexity of the DeepConvLSTM architecture. Assuming both layers in the original LSTM have the same number of hidden units h , the number of learnable parameters p_2 within the LSTM of DeepConvLSTM is $p_2 = 4sh + 8h + 12h^2$ with s denoting the size of the sliding window. By removing the second layer, the number of LSTM parameters in our architecture reduces to $p_1 = 4sh + 4h + 4h^2$, which shows that the increase in parameters with an increasing number of hidden units is dominated by $4h^2$ for p_1 opposed to $12h^2$ for p_2 . In our experiments the removal of one LSTM layer roughly equated in 63% fewer learnable LSTM parameters across all experiments. This decrease in complexity can be also seen in decreased runtimes using our architecture compared to the original DeepConvLSTM architecture. On average we are seeing a decrease in runtime of 30% across all experiments with the difference increasing (going up to as much as 48%) the more hidden units are employed and the larger the input dataset is.

4.4. Conclusion and Future Directions

We proposed to change the DeepConvLSTM architecture as frequently used in activity recognition to employ a 1-layered instead of a 2-layered LSTM. We validated our architecture change with experiments using 5 publicly available datasets [142, 149, 164, 140, 160] with varying numbers of hidden units within the LSTM.

Results show that for 4 out of 5 datasets, one LSTM layer consistently outperforms the original DeepConvLSTM architecture [126] in terms of precision, recall and F1-score. For [140], it outperforms the original 2-layer architecture only for LSTMs with smaller number of hidden units, and is on par for larger LSTMs. With our suggested removal of the second LSTM layer, we are able to decrease the number of learnable parameters of the LSTM within the DeepConvLSTM [126] by an average factor of 62% and are able to decrease training time on average by a factor of 38%, going up to as much as 48% for larger networks (with more hidden units).

These findings contradict with the belief that one needs at least a 2-layered LSTM when dealing with sequential data [89]. As this belief finds its origins in the area of text analytics,

we suggest that it might not be applicable to sensor-based HAR. Our choice of datasets to use for our experiments was driven by testing our hypotheses against datasets containing complex (Wetlab and Opportunity), transitional (SBHAR) and simpler/periodical activities (SBHAR, HHAR and RWHAR). Looking at the results one can see that the hypothesis holds for all three types of activities. Furthermore, with the Opportunity dataset and its 113 feature dimensions we showed that a performance increase can be also witnessed when using more than one sensor. Therefore we currently cannot clearly identify specific reasons for why 1-layered LSTMs are performing better within the setting of HAR. Next steps within this research include investigating whether the hypothesis holds true for larger HAR datasets to further rule out overfitting being a reason for the performance increase. Using e.g. a feature extraction tool as seen in [96] could be used to create such larger datasets. Further, we will analyse how 1-layered LSTMs learn compared to 2-layered ones within the setting of HAR by replicating a similar analysis as performed by Karparthy et al. [89].

5. Temporal Action Localization for Inertial-based Human Activity Recognition

Title: Temporal Action Localization for Inertial-based Human Activity Recognition

Authors: Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, Michael Moeller

Venue: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), Volume 8, Issue 4

Publication date: 21 November 2024

DOI: 10.1145/3699770

Code: github.com/mariusbock/tal_for_har

Building on insights from the benchmark analysis of the WEAR dataset (see Chapter 2), this work presents the first systematic application of vision-based TAL models to inertial-based activity recognition. It provides an extensive benchmark showing that TAL models consistently outperform traditional window-based HAR models across multiple datasets. By processing entire timelines rather than isolated windows, the work makes the case that TAL models could offer a compelling solution to the long-standing sliding window problem discussed in Chapter 1. TAL models offer many unexplored design choices and training concepts for a multitude of application scenarios, which could be of interest to the inertial-based community. Note that the article refers to a preprint version of the WEAR dataset [19], as both articles were published simultaneously.

Contextualization

Traditional inertial-based HAR approaches typically aim to classify fixed-size sliding windows with a single activity label. While this allows near-online applications of these models, it effectively introduces limitations in modeling temporal dependencies, as windows are treated individually by the classifier and temporal context does not extend across to previous or future sliding windows. While benchmarking the WEAR dataset (see Chapter 2) the TAL community was deemed to relate closest in their prediction scenario to that of the inertial activity recognition. Though both communities aim to localize activities within a timeline, the TAL community follow a segment-based rather than a per-window classification approach. Specifically, single-stage TAL models, as investigated as part of the benchmark analysis of

the WEAR dataset and work described in the following chapter, are designed to predict a timeline of activity segments, each defined by an activity label, start time and end time. To do so, these models process a sequence of sliding windows and extend standard classification of sliding windows, as e.g. done by the inertial community, with a regression task, in which each window predicts its relative position to the start (onset) and end (offset) of the activity segment it resides in. Finally, segment proposals and associated confidence scores of all windows are collected and used to reconstruct the activity timeline.

Originally developed for video data, TAL models had not been systematically applied to inertial modalities prior to this work. Initial experiments conducted as part of the WEAR dataset showed that such models could be effectively adapted to inertial signals (see Chapter 2). The following work expands on these findings by providing a comprehensive benchmark study across six widely used HAR benchmark datasets, revealing the superiority of single-stage TAL models, namely the ActionFormer [204], TemporalMaxer [167] and TriDet [151], compared to popular inertial-based architectures. By leveraging local and global context, the models showed to produce less fragmented activity timelines, handle complex and non-periodic activities more consistently, and achieve the highest NULL-class accuracies across all experiments. Additional experiments demonstrate that TAL models are effective even in short input sequences and that their performance improves further when using learned latent representations instead of raw inertial data. In conclusion, the paper argues that TAL models introduce a new paradigm with numerous unexplored design opportunities and training strategies that could significantly advance a wide range of HAR applications. This work further makes the case that TAL models could provide a solution to the sliding window problem of the inertial-based community discussed in Chapter 1, as reasoning over complete timelines of sliding windows allows models to reason via intra- and inter-window context, ultimately making hyperparameters chosen during segmentation less crucial.

The initial idea to apply TAL models to inertial data was explored by Marius Bock. While presenting and discussing early benchmarking results on the WEAR dataset (see Chapter 2), Michael Moeller proposed further pursuing TAL models for inertial-based HAR. Marius Bock conducted the literature review and comparison of TAL and inertial-based models. Kristof Van Laerhoven providing feedback on the selection of datasets for benchmarking the TAL models and the broader implications for the HAR community. All experiments were conducted and evaluated by Marius Bock, with Michael Moeller and Kristof Van Laerhoven offering feedback throughout the writing process to enhance the clarity and readability of the final article. This work was further refined through the feedback of anonymous *IMWUT* reviewers, who encouraged the inclusion of both raw inertial data and learned latent representations, and recommended testing the models in near-online settings. They also emphasized the significance of the work in addressing the sliding window problem. Their suggestions

helped further improve the experiments and contextualization of the study.

Abstract

As of today, state-of-the-art activity recognition from wearable sensors relies on algorithms being trained to classify fixed windows of data. In contrast, video-based Human Activity Recognition, known as TAL, has followed a segment-based prediction approach, localizing activity segments in a timeline of arbitrary length. This paper is the first to systematically demonstrate the applicability of state-of-the-art TAL models for both offline and near-online HAR using raw inertial data as well as pre-extracted latent features as input. Offline prediction results show that TAL models are able to outperform popular inertial models on a multitude of HAR benchmark datasets, with improvements reaching as much as 26% in F1-score. We show that by analyzing timelines as a whole, TAL models can produce more coherent segments and achieve higher NULL-class accuracy across all datasets. We demonstrate that TAL is less suited for the immediate classification of small-sized windows of data, yet offers an interesting perspective on inertial-based HAR – alleviating the need for fixed-size windows and enabling algorithms to recognize activities of arbitrary length. With design choices and training concepts yet to be explored, we argue that TAL architectures could be of significant value to the inertial-based HAR community. The code and data download to reproduce experiments is publicly available via github.com/mariusbock/tal_for_har.

5.1. Introduction

The recognition of performed activities through wearable sensors such as IMUs has shown to be of significant value in areas such as health care or the support of complex work processes [26]. With the works of many researchers exploring lightweight architectures [210], much of the success of inertial-based models has stemmed from applying them in an online-fashion on wearable edge devices [137, 182]. As shown to work on a multitude of HAR-related scenarios, it has established itself within the inertial-based community to employ a window-based prediction approach using a predefined window size and overlap. With classification algorithms being tasked to assign a label to each window individually, this approach has enabled inertial-based architectures to classify newly incoming data at will. As a too large window size may include multiple fast actions within a single window, and a too short window may not be enough to capture a complete lengthier action, the chosen size of the fixed window has become a crucial parameter within traditional recognition systems.

A persistent trend in Deep Learning has been the applicability of machine learning concepts such as self-attention [177] to other areas and application scenarios than originally

introduced for. With significant progress having been made since the introduction of deep neural architectures such as DeepConvLSTM [126], researchers have followed this trend and have continuously worked on improving the architectural design of networks by incorporating newly introduced techniques (see, e.g., [210]). A promising recent approach in video-based HAR is TAL, which aims to locate activity segments, defined by a class label, start, and end point, within an untrimmed video. Even though introduced architectures have almost doubled in performance over the last 5 years on existing datasets like THUMOS-14 [88], results on large corpora such as EPIC-KITCHENS-100 [38] and Ego4D [59] show that the prediction problem is far from being saturated.

Despite sharing a mutual goal, the TAL community as opposed to the inertial community does not follow an online-based prediction approach, but aims to classify recorded times sequences in an offline fashion, taking advantage of learnable temporal dependencies along the complete timeline. Furthermore, by following a segment-based prediction with activity boundaries being learned via regression-based loss, the "window size problem" is alleviated with methods being flexible enough to recognize both short and long lasting activities. TAL models have recently been shown to be capable of being trained using raw inertial data [19], marking the first instance of such vision-based models being applied in the context of inertial-based HAR. When compared with popular inertial models, results on one particular dataset have already shown that TAL models can produce more coherent and less fragmented predictions while maintaining performance in terms of traditional classification metrics. In light of activity recognition systems having also been deployed to provide offline analysis of streams of prerecorded data [83, 176, 40, 14, 203, 12], this work sets out to further examine the applicability of Temporal Action Localization for inertial-based HAR. Our contributions in this paper are three-fold:

1. We demonstrate the capabilities of a novel approach to inertial-based HAR, being single-stage TAL, to outperform popular, window-based inertial models on a multitude of wearable activity recognition benchmark datasets in an offline-prediction scenario.
2. Complementing traditional, window-wise classification metrics, we introduce a set of unexplored, segment-based evaluation metrics for inertial-based HAR, which are based around the scalar evaluation metric mean Average Precision [6].
3. Though our results demonstrate the superiority of inertial-based models in an online prediction scenario, we show that TAL models can be applied in a near-online fashion, functioning e.g. a server-side prediction tool which provides prediction with a lag of 1 minute.

5.2. Related Work

Inertial-based Human Activity Recognition With on-body sensors providing a robust and non-intrusive way to monitor participants along long stretches of time, research conducted in the area of inertial-based HAR has worked towards the automatic interpretation of one or multiple sensor streams to reliably detect activities e.g. in the context of providing medical support or providing guidance during complex work processes [26]. With deep neural networks having established themselves as the de facto standard in inertial-based HAR, DeepConvLSTM [126] as well as the models introduced at a later point follow a similar prediction scenario design applying a sliding window approach which groups concurrent data points for classification. In their original publication Ordóñez and Roggen [126] have found a combination of convolutional and recurrent layers to produce competitive results, with the idea being to model temporal dependencies amongst automatically extracted discriminative features within a sliding window in order to classify it correctly as one of the N activity classes or, if applicable, NULL-class. Building on the idea of combining these two types of layers, researchers have worked on extending the original DeepConvLSTM or have introduced their own architectural designs [121, 174, 131, 188, 191, 189, 108, 34, 17, 1, 210, 205, 119, 33, 32, 186]. A simple modification of the DeepConvLSTM is the shallow DeepConvLSTM [17]. Contradicting the popular belief that one needs at least two recurrent layers when dealing with time series data [89], Bock et al. [17] demonstrated that removing the second LSTM layer within the original DeepConvLSTM architecture results in significant increases in predictive performance on a multitude of HAR benchmark datasets while also decreasing the number of learnable parameters. Furthermore, with the original DeepConvLSTM only being able to learn temporal dependencies within a sliding window, the shallow DeepConvLSTM applies the LSTM across batches, effectively making the batches the sequence which is to be learned by the LSTM. This dimension flip, along with a non-shuffled training dataset, enables the architecture to learn temporal dependencies amongst a batched input. The same year as the shallow DeepConvLSTM, Abedin et al. [1] introduced Attend-and-Discriminate, a deep neural network architecture following the idea of the original DeepConvLSTM by combining both convolutional and recurrent layers, yet further adding a cross-channel interaction encoder using self-attention as well as attention mechanism to the recurrent parts of the network. In 2022 Zhou et al. [210] proposed TinyHAR, a lightweight HAR model that uses a transformer encoder block along with means of cross-channel fusion via a fully connected layer and a final self-attention layer which aims to learn the contribution of each outputted time step produced by the recurrent parts individually.

Video-based Human Activity Recognition Classifying videos in the context of Human Activity Recognition can be broadly categorized into three main application scenarios: Action Recognition, which aims to classify trimmed videos into one of C activity classes; Action Anticipation, which aims to predict the next likely activities after observing a preceding video sequence; and TAL, which seeks to identify and locate all activity segments within an untrimmed video. With the inertial-based benchmark datasets consisting of a multitude of activities, TAL is to be considered most comparable to inertial-based HAR. Unlike popular inertial architectures though, TAL models aim to predict all segments within an untrimmed video at once. Existing TAL methods can broadly be categorized into two categories: two-stage and one-stage recognition systems. Two-stage recognition system [106, 103, 193, 9, 208, 202, 58, 110, 134, 156, 212, 207, 165] divide the process of identifying actions within an untrimmed video into two subtasks. First, during proposal generation, candidate segments are generated, which are then, during the second step, classified into one of C activity classes and iteratively refined regarding their start and end times. Contrarily, single-stage models [195, 152, 122, 111, 107, 114, 104, 30, 204, 167, 151] do not rely on activity proposals, directly aiming to localize segments along their class label and refined boundaries. In 2022 Zhang et al. [204] introduced the ActionFormer, a lightweight, single-stage TAL model which unlike previously introduced single-stage architectures does not rely on pre-defined anchor windows. In line with the success of transformers in other research fields, Zhang et al. [204] demonstrated their applicability for TAL, outperforming previously held benchmarks on several TAL datasets [71, 38, 88] by a significant margin. Surprisingly, a year later, TemporalMaxer suggested removing transformer layers within the ActionFormer, arguing that feature embeddings are already discriminative enough [167]. Though being more lightweight than the ActionFormer, the TemporalMaxer showed to outperform its precedent during benchmark analysis. Similarly to the TemporalMaxer, Shi et al. [151] introduced TriDet, which suggested altering ActionFormer in two ways. First, to mitigate the risk of rank-loss, self-attention layers are replaced with SGP layers. Second, the regression head in the decoder is replaced with a Trident head, which improves imprecise boundary predictions via an estimated relative probability distribution around each timestamp’s activity boundaries.

5.3. Temporal Action Localization for Inertial-based HAR

As the inertial-based HAR and TAL communities deal with inherently different modalities, both communities have developed distinct predictive pipelines and algorithms tailored to the challenges and characteristics of their respective modalities (see Figure 5.1). The objective of both inertial activity recognition and TAL is to predict all activities present in an untrimmed

timeline. Given an input data stream X of a sample participant, both the inertial and TAL communities start by applying a sliding window approach which shifts over X , dividing the input data into windows, e.g. of one second duration with a 50% overlap between consecutive windows. This process results in $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ being discretized into $t = \{0, 1, \dots, T\}$ time steps, where T is the number of windows for each participant. It is important to note that the TAL models do not use raw data as input but are instead trained using feature embeddings extracted from each individual sliding video clip, which are extracted using pre-trained methods such as [27].

Given all sliding windows associated with an untrimmed sequence, inertial activity recognition models aim to predict an activity label a_t for each sliding window $\mathbf{x}_t$, where a belongs to a predefined set of activity labels, $a_t \in 1, \dots, C$. To do so, the sliding windows are batched and fed through, e.g., a deep neural network, such as the DeepConvLSTM [126]. The resulting activity labels for each window are then compared to the true activity labels from the ground truth data, and classification metrics like accuracy or F1-score are calculated. Contrarily, TAL models aim to identify and localize segments of actions within the untrimmed data stream, which can span across multiple windows. To achieve this algorithms are trained to predict activity segments $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, where N varies across participants. Each activity instance $\mathbf{y}_i = (s_i, e_i, a_i)$ is defined by its starting time s_i (onset), end time e_i (offset) and its associated activity label a_i , where $s_i \in [0, T]$, $e_i \in [0, T]$, and $a_i \in \{1, \dots, C\}$.

As the input data used to train TAL models is a collection of 1-dimensional feature embeddings, the 2 d-dimensional, windowed inertial data commonly found in the inertial-based HAR community, cannot directly be used to train TAL models. The following will thus describe two preprocessing methods which can be used to convert the inertial data into a format such that it can be used as input for TAL models.

Vectorization of raw inertial data Since both communities employ a sliding window approach but feed data to their models using different dimensions, Bock et al. [19] proposed a simple, yet effective preprocessing step. This step converts the 2-dimensional, windowed inertial data, as used by inertial architectures, into a 1-dimensional feature embedding suitable for training TAL models. The preprocessing method as detailed in Figure 5.2 involves concatenating the different sensor axes of each window, converting the input data to be a collection of 1-dimensional feature embedding vectors. More specifically, given a sliding

Inertial Activity Recognition:

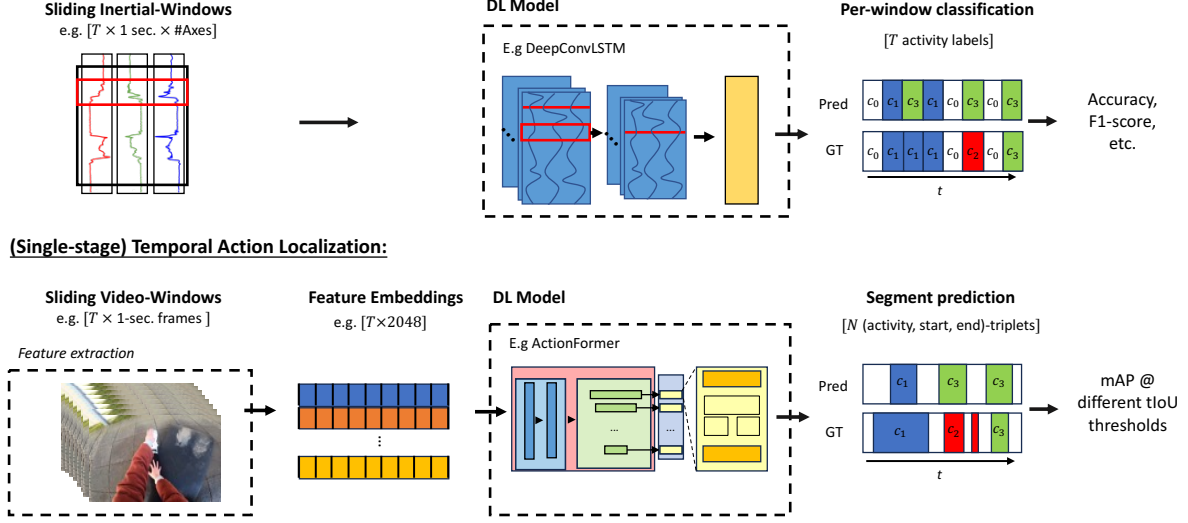


Figure 5.1.: Overview of the prediction pipelines applied in inertial-based activity recognition and single-stage TAL. Both apply a sliding window to divide input data into windows of a certain duration (e.g. one second). TAL models do not use raw data as input but are applied on per-clip, pre-extracted feature embeddings. Inertial activity recognition models predict activity labels for each sliding window, which are used for calculating classification metrics such as accuracy and F1. TAL models predict activity segments, defined by a label, start and end points, and are evaluated with mean Average Precision (mAP) applied at different tIoU thresholds.

window $\mathbf{x}_{SW} \in \mathbb{R}^{W \times S}$, we vectorize the two-dimensional matrix as follows:

$$\mathbf{x}_{SW} = \begin{bmatrix} x_{11} & \dots & x_{1S} \\ \vdots & \ddots & \vdots \\ x_{W1} & & x_{WS} \end{bmatrix} \rightarrow \text{vec}(\mathbf{x}_{SW}) = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1S} \\ x_{2S} \\ \vdots \\ x_{WS} \end{bmatrix} \quad (5.1)$$

Two-stage training via prepended inertial models In order to encode videos and come up with a discriminative, latent feature representation, TAL models usually resort to using extracted feature embeddings from models pretrained on large vision corpuses like Kinetics-400 [90]. Inspired by this, we propose a second variant on how to use inertial data as input to TAL models (see Figure 5.3) which involves using features extracted from a separately trained inertial model as latent representation of each sliding window. Specifically, a LOSO cross-validation step within the two stage training consists of 1. training the inertial model on

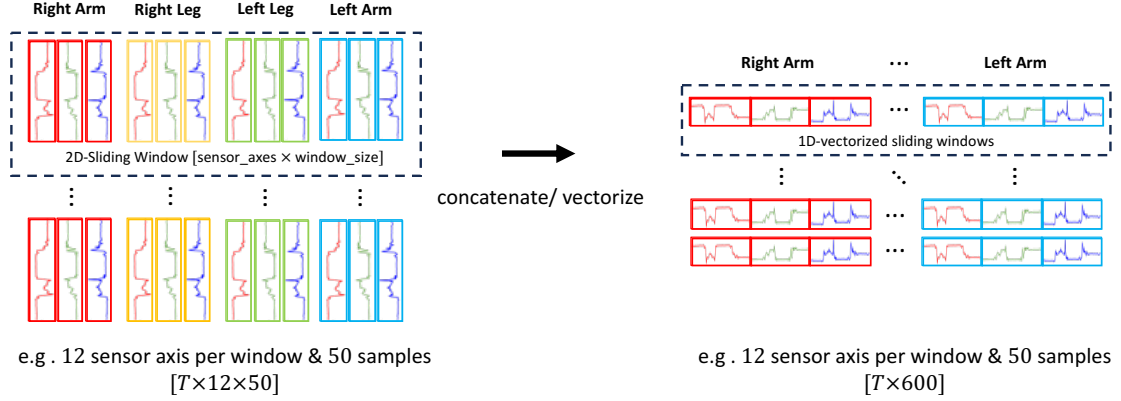


Figure 5.2.: Visualization of the applied vectorization on top of windowed inertial data assuming four 3D-inertial sensors and a sliding window size of 50 samples. Each 2D-sliding-window of size $[50 \times 12]$ is vectorized by concatenating each of the axes one after another. Resulting 1D-embedding vectors, being of size $[1 \times 600]$, can be used to train TAL models.

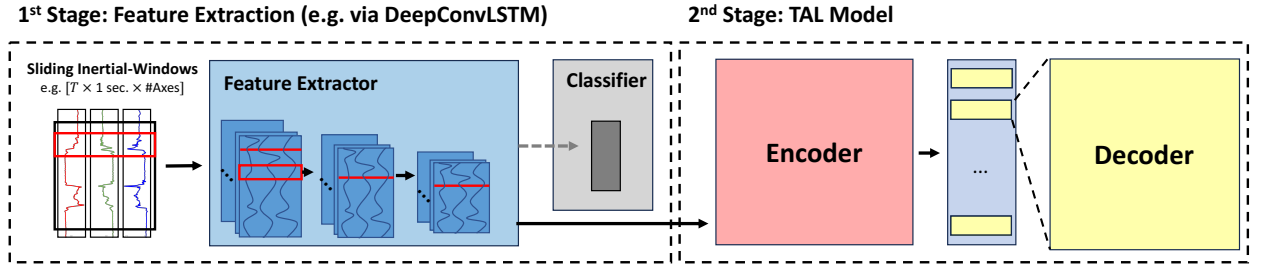


Figure 5.3.: Visualization of the applied two-stage training process. The first stage involves training e.g. a classic DeepConvLSTM as introduced by Ordóñez and Roggen [126]. Once the first-stage training has finished, the classifier is omitted from the model such that latent features can be extracted. The 1-dimensional, window-wise features are then used as input embeddings for the second stage, i.e. training a TAL model.

all but the validation data, 2. use the trained inertial network to extract latent feature representations of each window within the training and validation data, and 3. using the extracted features as clip-wise feature embeddings to train a TAL model.

5.3.1. TAL Architectures Overview

In light of the recent success of transformer-based models in e.g. Natural Language Processing and Computer Vision, Zhang et al. [204] proposed the *ActionFormer*, is an end-to-end trainable transformer-based TAL architecture. Unlike other single-stage TAL approaches, it does not rely on pre-defined anchor windows. The architecture, as illustrated in Figure 5.4, combines multiscale feature representations with local self-attention and is trained through a classification and regression loss calculated by a light-weighted decoder. Building up on the ActionFormer architecture, Tang et al. [167] and Shi et al. [151] proposed the

TemporalMaxer and *TriDet* model respectively. Within the *TriDet* model projection and transformer layers of the ActionFormer are replaced with fully-convolutional SGP layers and the regression head is replaced by a trident regression head which claims to improve imprecise boundary predictions. Contrarily, the TemporalMaxer suggests modifying the encoder of the ActionFormer to employ solely max pooling and remove all transformer-based layers, as, according to the authors, this does not come at the cost of a lost in information and predictive performance.

In order to predict activity segments $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ within an input video, the ActionFormer, TemporalMaxer and TriDet model all follow the same sequence labeling problem formulation for action localization. That is, given a set of one-dimensional feature input vectors $\mathcal{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_T\}$ extracted from $\mathcal{X}$, a model aims to classify each timestamp as either one of the activity categories C or as background (or null) class and regress the distance towards the timestamp's corresponding segment's start and end point. More specifically, given the input vectors $\mathcal{X}'$ a model aims to learn to label the sequence as

$$\mathcal{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_T\} \rightarrow \hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_T\}, \quad (5.2)$$

where $\hat{\mathbf{y}}_t = (p(a_t), d_t^s, d_t^e)$ at timestamp t is defined by a probability vector $p(a_t)$ indicating the class-wise probability of the timestamp being classified as one of the activity categories C and $d_t^s > 0$ and $d_t^e > 0$ being the distance between the current timestamp t and the current segment's onset and offset. Note that d_t^s and d_t^e are not defined if the timestamp is to be classified as background. The sequence labeling formulation can then be easily decoded to activity segments with:

$$a_t = \arg \max p(a_t), \quad s_t = t - d_t^s, \quad \text{and} \quad e_t = t + d_t^e \quad (5.3)$$

The authors of the ActionFormer, TemporalMaxer, and TriDet models attribute much of their models' performance to the constructed multi-layer feature pyramid. The feature pyramid within each of the three models downsamples the input sequence of sliding windows multiple times to create representations of a participant's data stream at different temporal granularities [204, 167, 151]. Using this method, the TAL models can learn both short and long temporal dependencies across a participants's timeline, as the temporal distance between two embedding vectors within the feature pyramid increases the further we move down the pyramid. This use of downsampling to capture different length temporal patterns is comparable to how inertial models concatenate convolutional layers without padding to increase the receptive field of each convolutional kernel. However, both types of models differ in the types of patterns learned. The feature pyramid of the TAL models enables them to

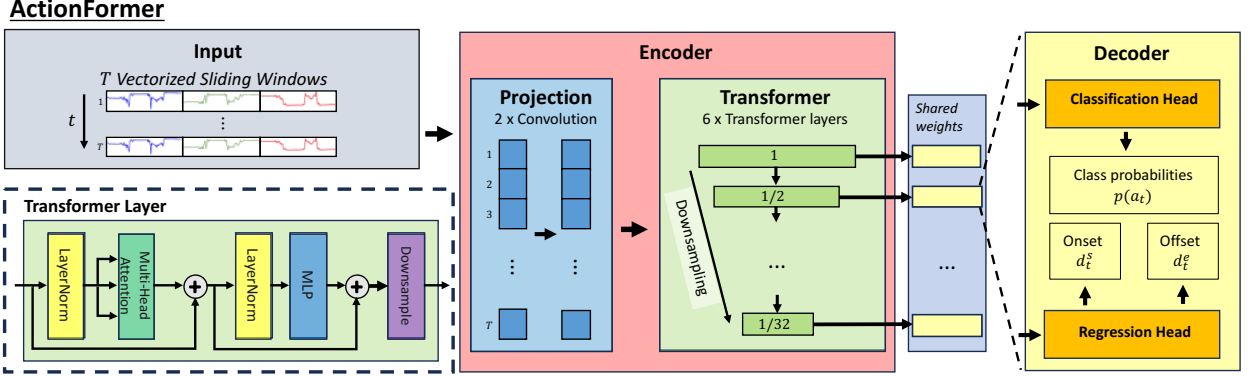


Figure 5.4.: Architecture overview of the ActionFormer proposed by Zhang et al. [204]. The architecture follows an encoder-decoder structure. The encoder encodes input sequences into a feature pyramid, which captures information at various temporal scales. The decoder, consisting of a classification and regression head, then decodes each timestamp within the feature pyramid to sequence labels, i.e. a class probability vector and the timestamp’s activity onset and offset distance. The TriDet[151] and TemporalMaxer[167] both follow the same encoder-decoder structure, yet suggest architectural changes.

learn cross-window, temporal patterns of arbitrary length, while the convolutional parts of the inertial models enable them to learn in-window temporal patterns of arbitrary length.

Due to the TAL community’s training objective differing from that of the inertial community, TAL models are trained not only on a classification loss, but also a regression loss, which optimizes each timestamp’s corresponding activity onset and offset. The use of self-attention based layers has shown to improve results in both communities, yet variations of the ActionFormer show that these layers are not necessarily needed. More specifically, the use of fully-convolutional SGP layers in the TriDet model show that, rather than the type of technique being important, focus needs to be put specifically on learning both local and global temporal information for each timestamp.

5.4. Methodology

5.4.1. Datasets

We evaluate each algorithm featured in this benchmark analysis using 6 popular HAR datasets, namely the Opportunity [142], SBHAR [140], Wetlab [149], WEAR [19], Hang-Time [77] and RWTHAR dataset [164]. The datasets, all covering different application and recording scenarios, provide us with a challenging set of prediction problems to properly assess the strengths and weaknesses of each model. Table 5.1 summarizes the key characteristics of each dataset. In addition to vital information such as participant count, activity count, sampling rate and sensor axes count, the table also includes details on the scenario and type of activities found in each dataset. We classify activities into four types: (1) pe-

Table 5.1.: Key characteristics of the datasets used in this benchmark analysis. The table provides: participant count (#Sbjs), activity count (#Cls), sampling rate (SR), sensor axes count (#Axes), average number of segments per participant based on absolute length of the segment (#Segments), overall scenario (e.g. activities of daily living (ADL)) and type of activities found in the dataset.

Dataset	#Sbjs	#Cls	SR	#Axes	#Segments [6]					Scenario	Type of Activities
					XS	S	M	L	XL		
Opportunity [142]	4	17(+1)	30	113	403 ± 63	185 ± 48	48 ± 10	3 ± 2	0 ± 0	ADL	non-periodic, complex
SBHAR [140]	30	12(+1)	50	3	3 ± 2	9 ± 2	6 ± 4	10 ± 4	13 ± 3	Locomotion	(non-)periodic
Wetlab [149]	22	8(+1)	50	3	7 ± 5	7 ± 3	7 ± 2	5 ± 2	13 ± 2	Laboratory	complex
WEAR [19]	18	18(+1)	50	12	1 ± 1	1 ± 1	1 ± 1	1 ± 1	30 ± 9	Sports	periodic
Hang-Time [77]	24	5(+1)	50	3	154 ± 53	22 ± 7	10 ± 7	3 ± 3	3 ± 2	Sports	(non-)periodic, complex
RWHAR [164]	15	8	21	50	0 ± 0	0 ± 0	0 ± 0	0 ± 0	9 ± 1	Locomotion	periodic

riodic activities, characterized by recurring periodic patterns; (2) non-periodical activities consisting of non-occurring, non-periodical patterns; and (3) complex activities, defined by an arbitrary sequence of non-periodic and periodic activities. Lastly, following the works of Alwassel et al. [6], we provide average count of segments across all subjects, categorizing each segment within each dataset into five bins: XS: (0 seconds, 3 seconds], S: (3 seconds, 6 seconds], M: (6 seconds, 12 seconds], L: (12 seconds, 18 seconds], and XL: more than 18 seconds.

5.4.2. Training Pipeline

Prediction scenarios All experiments were initiated employing a Leave-One-Subject-Out cross-validation. This type of validation involves iteratively training on all but one participant’s data and using the hold-out participant’s data during validation until all participants have been evaluated, ensuring that each network architecture is assessed based on its capabilities to generalize across unseen participants. All datasets were windowed using a sliding window of one second with a 50% overlap across windows. In order to allow for fair comparison, our benchmark analysis assesses the TAL models employing two prediction scenarios, with the differences in their predicted output between inertial-based and TAL models. Being the scenario TAL models were intended to be used for, within the first prediction scenario (*Offline Activity Recognition*) the TAL models are tasked to predict each participant’s data as a whole, i.e. one batch representing the full data stream available of one participant. Nevertheless, to also assess the models’ capabilities to provide window-level predictions, within the second prediction scenario (*Online Activity Recognition*) we artificially chunk participants’ timelines making the models predict the timelines in chunks (e.g., 30 seconds). As inertial-based models unlike TAL models are designed to make window-based predictions, the second prediction scenario will assess the TAL models online capabilities while also benchmarking them with removed temporal context. Note that in case of the *Offline Activity Recognition* we test the TAL models using both a single-stage as well as

two-stage training (as described in Chapter 5.3).

Hyperparameters For all inertial architectures [126, 17, 1, 210] we employed a similar optimization as proposed with the release of the shallow DeepConvLSTM, namely an Adam optimizer paired with a learning rate of $1e^{-4}$, a weight decay of $1e^{-6}$, and Glorot weight initialization [57]. To allow each model to converge more properly, we increase the number of epochs to 100 and employ a step-wise learning-rate schedule that multiplies the learning rate by a factor of 0.9 after every 10 epochs. For all architectures, we fixed the hidden dimension of the recurrent layers to employ 128 units and scaled kernel sizes of the convolutional filters according to the relative difference in sampling rate among the different input datasets. In line with how the Attend-and-Discriminate architecture was first introduced, we optimized said architecture using center-loss [185] as opposed to a weighted cross-entropy loss, which was used during training of all other inertial architectures. Lastly, as proposed by the authors, we do not shuffle batches during the training of the shallow DeepConvLSTM.

The three TAL architectures [204, 167, 151], we chose to employ hyperparameters reported by the authors that produced best results on the EPIC-Kitchens dataset [38]. The hyperparameters, though optimized for a different modality than inertial data (egocentric videos), have shown to produce competitive results on the WEAR dataset [19] and are thus considered a good starting point for evaluating the applicability of the three architectures on other HAR datasets. Nevertheless, given the small size of the tested inertial datasets compared to datasets used by the TAL community, we chose to increase the amount of epochs to 100 throughout all TAL-based experiments. As TAL models are designed to predict both a regression (boundary prediction) and classification target (segment label), loss calculation of the ActionFormer, TemporalMaxer and TriDet models are performed as a weighted combination of a regression loss (IoU loss [141]) and classification loss (focal loss [100]).

Postprocessing To compare predictions of both TAL and inertial-based models, segments (TAL) and windowed-predictions (inertial-based) first needed to be translated back to sample-wise predictions. In case of the segments we do so by translating each segment "overwrite" the prediction timeline with its associated label. By starting with the segment associated with the lowest prediction confidence, segments with a high prediction confidence are preferred in case of overlapping segments. In case of the inertial-based architectures, we start by iterating over the windowed predictions in order of occurrence in the timeline having them determine the samples they are associated with. To deal with the overlap amongst windows, preceding windows are allowed to "overwrite" the label of samples of previous windows which they are overlapping with.

As inertial models are tasked to predict on a per-window basis and, the architectures suffer

from frequently occurring activity switches ultimately leading to fragmented segments which significantly lower mAP scores being produced as opposed to the TAL models. Therefore, to remove only short lasting switches, predictions of inertial-based architectures mentioned in this paper were smoothed using majority-vote filters. The exact size of the majority filter was chosen dataset-specific, determined via trying out a selection of filters between 2.5 and 40 seconds. Specifically, the filters were chosen as: 2.5 seconds (Opportunity), 20 seconds (Wetlab), 5 seconds (SBHAR), 15 seconds (WEAR), 5 seconds (Hang-Time) and 40 seconds (RWHAR). Similar to the inertial models, the TAL architectures are tasked to predict class probabilities and segment boundaries of each windowed timestamp. Consequently, without applying any confidence threshold, all predicted activity segments are considered during creation of the prediction timeline causing accuracy of the NULL-class to be significantly low. Therefore, to alleviate this, we apply an optimized confidence threshold on predicted segments of all TAL models. Similarly to the majority filter, the score threshold for each architecture was chosen dataset-specific, determined via trying out a selection of thresholds between 0.05 and 0.5 seconds. This eliminates low scoring segments and substantially lowers the confusion of the NULL-class with the other activities. More details on the effect of the majority filter as well as score thresholding can be found in the supplementary material.

5.5. Results

As part of our experimental evaluation, we provide traditional classification metrics (precision, recall and F1-score), misalignment measures as defined by Ward et al. [182] and mAP averaged across tIoU thresholds 0.3, 0.4, 0.5, 0.6 and 0.7. All results are the unweighted average across all subjects along the LOSO validation. Experiments were repeated three times employing three different random seeds (1, 2 and 3). Classification metrics are calculated on a per-sample basis as the segmented output of the TAL models and windowed output of the inertial-models need to be translated to a common time granularity. To ensure readability of this work, visualization of the per-class analysis will only include confusion matrices of the SBHAR and RWHAR datasets, as we deemed these two datasets to be the most representative in illustrating the strengths and weaknesses of the TAL architectures when applied to inertial data. Please note that the confusion matrices of the other datasets can be found in the supplementary material. Furthermore, all created plots part of a performed DETAD analysis [6] on each dataset can be found in our repository.

5.5.1. Offline Activity Recognition

Table 5.2 provides average results of the seven tested architectures across each dataset in an offline prediction setting. One can see that the TAL architectures outperform the inertial

architectures across all datasets regarding average mAP. This shows that by being trained to specifically optimize activity boundaries, the different prediction target has resulted in overall more coherent segments which overlap to a larger degree with the ground truth segments. Even though prediction results of the inertial architectures were further smoothed by a majority filter, average mAP is, except for the WEAR dataset, more than halved when compared to that of the TAL architectures. Regarding traditional classification metrics the TAL architectures are able to outperform inertial architectures for four out of six datasets with only the RWHAR [164] and WEAR dataset [19] being the exception. These improvements range between 5% for the Opportunity and Wetlab dataset, 10% for the Hang-Time and even as much as 25% in F1-score for the SBHAR dataset. Calculated misalignment ratios show that both inertial and TAL architectures have a similar distribution of errors, with architectures producing better overall classification results also showing overall lower misalignment measures. Only for the RWHAR dataset one can see that TAL architectures show a significantly higher Overfill-Ratio. This might be due to the RWHAR dataset not featuring a NULL-class, which introduces an uncommon prediction scenario for the TAL architectures. Nevertheless, the performed DETAD analysis [6], which further differentiates amongst the segment-based errors, reveals that inertial architectures suffer more severely from background errors, i.e. confusing activities with the NULL-class. While this effect is decreased for the shallow DeepConvLSTM, TAL architectures show to be able to more reliably differentiate between activities and the NULL-class, and thus more reliably localize activities within the untrimmed sequences.

With the improvements on the SBHAR being the largest across all datasets, one can see on a per class level (see Figure 5.5) that this increase can be attributed to improved performance on transitional, non-periodic classes like *sit-to-stand*. These activities are mostly recognisable by their context and surrounding activities and are thus particularly challenging to predict for models that do not rely on temporal dependencies spanning multiple seconds. By applying the TAL architectures in an offline manner, the architectures are able to leverage both local and global context across the whole timeline and are thus even able to recognize these short-lasting activities. Since the DeepConvLSTM, Attend-and-Discriminate and TinyHAR models are being trained on shuffled training data and have recurrent parts applied on within-window sequences, the three architectures cannot learn to predict these activities based on surrounding context. Among inertial models, only the shallow DeepConvLSTM is able to more reliably predict these context-based classes as it is trained on unshuffled data and applies a dimensional flip when training the LSTM. Across all datasets part of this benchmark analysis, the RWHAR dataset yields the least performant results for the TAL architectures. We accredit this primarily due to the absence of a NULL-class in the dataset, which introduces a uncommon prediction scenario for the models. Furthermore,

Table 5.2.: Offline Activity Recognition: Average LOSO cross-validation results obtained on six inertial HAR benchmark datasets [142, 140, 149, 19, 77, 164] for four inertial [126, 17, 1, 210] and three TAL architectures [204, 167, 151]. The table provides per-sample classification metrics, i.e. Precision (P), Recall (R), F1-Score (F1), misalignment ratios [182] and average mAP applied at different tIoU thresholds (0.3:0.1:0.7). All experiments employed a sliding window of one second with a 50% overlap. Results are averaged across three runs employing different random seeds. The TAL architectures are able to outperform the inertial architectures regarding average mAP on all HAR datasets and result in the highest classification metrics on four out of the six datasets. Best results per dataset are in **bold**.

	Model	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	mAP (↑)
Opportunity	DeepConvLSTM	50.22	33.88	34.41	0.30	17.29	0.78	31.26	0.01	0.23	13.97
	Shallow D.	42.08	27.18	26.46	0.24	14.28	0.97	35.06	0.01	0.15	10.61
	A-and-D	35.25	48.55	36.35	0.32	15.28	0.45	52.55	0.02	0.35	13.75
	TinyHAR	48.09	54.27	47.09	0.34	19.99	0.39	34.01	0.02	0.46	19.78
	ActionFormer	54.63	58.78	51.93	0.19	13.34	0.41	33.82	0.01	0.42	51.24
	TemporalMaxer	44.44	55.63	44.74	0.21	14.13	0.40	43.78	0.01	0.55	46.31
	TriDet	48.72	57.69	48.79	0.23	13.20	0.39	40.07	0.01	0.58	49.70
SBHAR	DeepConvLSTM	67.54	63.72	62.31	0.41	7.19	0.59	21.44	0.07	0.10	49.60
	Shallow D.	72.98	75.41	71.13	0.60	10.19	0.46	14.23	0.02	0.09	65.15
	A-and-D	68.64	71.07	66.49	0.31	9.47	0.45	21.71	0.05	0.11	55.79
	TinyHAR	58.91	63.70	56.29	0.31	8.16	0.54	31.67	0.05	0.13	45.38
	ActionFormer	87.02	84.37	84.43	0.36	5.41	0.16	5.09	0.00	0.20	95.46
	TemporalMaxer	86.52	83.37	83.66	0.41	6.02	0.19	4.41	0.00	0.24	94.39
	TriDet	88.95	86.15	86.45	0.38	5.41	0.12	3.95	0.01	0.29	94.75
Wetlab	DeepConvLSTM	38.65	47.34	37.87	0.51	7.13	0.69	48.53	0.21	0.64	11.88
	Shallow D.	39.01	35.42	34.42	0.51	9.52	1.57	34.51	0.06	0.43	15.40
	A-and-D	37.75	55.71	37.49	0.56	9.69	0.63	57.60	0.16	0.57	12.27
	TinyHAR	34.31	50.84	31.48	0.61	8.41	1.00	61.26	0.11	0.59	10.05
	ActionFormer	40.71	49.25	40.71	0.56	9.41	0.79	52.44	0.07	0.79	33.53
	TemporalMaxer	50.43	36.65	37.09	0.59	9.37	0.83	53.59	0.10	0.61	35.72
	TriDet	44.13	49.15	42.85	0.58	8.85	0.75	48.63	0.10	0.84	34.05
WEAR	DeepConvLSTM	80.68	76.25	75.78	0.28	2.35	0.52	6.68	0.11	0.32	61.03
	Shallow D.	80.78	78.91	77.71	0.27	3.23	0.50	5.21	0.04	0.43	67.89
	A-and-D	82.34	83.29	80.61	0.20	4.03	0.33	7.18	0.09	0.52	64.78
	TinyHAR	81.87	84.23	80.56	0.21	4.77	0.27	9.54	0.10	0.55	63.33
	ActionFormer	71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
	TemporalMaxer	69.54	72.80	69.52	0.23	6.87	0.83	5.50	0.01	1.50	69.18
	TriDet	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
Hang-Time	DeepConvLSTM	44.13	33.95	35.25	0.28	10.60	0.88	20.16	0.27	1.46	5.44
	Shallow D.	37.97	38.19	36.85	0.35	14.00	1.07	36.38	0.21	2.33	5.00
	A-and-D	40.54	43.32	40.39	0.35	14.14	0.72	34.81	0.30	1.44	6.73
	TinyHAR	37.09	41.13	36.89	0.41	11.59	0.75	42.90	0.43	1.26	4.73
	ActionFormer	49.19	57.57	51.23	0.62	11.63	0.51	47.65	0.48	0.64	29.26
	TemporalMaxer	45.01	54.56	47.17	0.71	10.97	0.45	52.71	1.13	0.65	27.86
	TriDet	49.59	55.14	50.67	0.73	9.85	0.52	48.55	0.69	0.62	29.24
RWHAR	DeepConvLSTM	79.05	81.93	77.56	0.65	2.05	1.09	13.63	1.07	0.00	0.11
	Shallow D.	88.59	89.01	86.85	0.31	1.14	0.38	6.93	0.98	0.00	0.00
	A-and-D	79.46	82.68	78.09	0.66	2.31	1.17	11.28	0.84	0.00	0.04
	TinyHAR	83.59	86.25	82.62	0.53	1.14	0.76	11.65	0.89	0.00	0.02
	ActionFormer	63.76	67.64	61.24	2.48	11.12	2.06	11.23	0.09	0.00	65.40
	TemporalMaxer	63.20	67.60	60.59	2.59	11.95	1.81	13.96	0.36	0.05	50.60
	TriDet	69.27	73.04	67.86	1.48	6.88	2.03	10.24	0.23	0.00	69.98

the RWHAR consists of the least amount of segments per subject, limiting the amount of training segments which can be used to optimize the TAL models. Results on the RWHAR dataset are nevertheless surprising as the TAL models show confusion among classes which they do not struggle to predict in other datasets (e.g., *lying* in the SBHAR dataset) as well as classes that are not similar to each other (e.g., *lying* and *jumping*). The obtained results raise the question of whether TAL models are primarily suited for being applied on untrimmed sequences, which (1) include breaks and/ or (2) provide a larger amount of segments than the RWHAR dataset.

Nevertheless, apart from the RWHAR dataset, the TAL models deliver the most consistent results across all classes. Even though, in most cases, the individual per-class accuracies are not the highest when compared to results obtained using the inertial architectures, the inertial architectures are frequently not able to predict all classes reliably, with at least one class showing low prediction accuracy. This is especially evident when examining the results obtained on the Hang-Time and Opportunity datasets, where TAL models are capable of correctly predicting challenging non-periodic activities such as *rebounds*, *passes*, *opening door* and *closing door*. Furthermore, as also seen in the DETAD analysis, TAL models are overall more capable of differentiating activities from the NULL-class, showing the highest NULL-class accuracy across all datasets. To summarize, the TAL architectures are more reliable in recognizing any kind of actions within an untrimmed sequence, and are less prone to predict fragmented prediction streams or non-existent breaks.

Single- vs. two-stage TAL training As described in Section 5.3, the TAL models described in this paper were intended to be applied using feature embeddings describing each sliding window, rather than raw data. Nevertheless, our initial results show that TAL models are indeed capable of being applied to raw inertial data. Figure 5.6 presents results of our implemented two-stage TAL training, which extends the training process as described in the previous chapters with a prepended feature extraction using inertial models. In total, we assess two feature embeddings: LSTM-based features extracted from a DeepConvLSTM [126] and attention-based features from a TinyHAR architecture [210]. Though the two-stage training using LSTM-based features only yields better F1-scores for the Wetlab, WEAR, and RWHAR datasets, the training using attention-based features improves results across all datasets across all TAL models. Given the significant improvements on the RWHAR and WEAR datasets, we assume that the feature extraction via the prepended inertial network helps increase the discriminability of the window-level features, yet at the cost of making it harder for the TAL models to learn cross-window temporal relations, as evident by the decrease in mAP scores across (almost) all datasets.

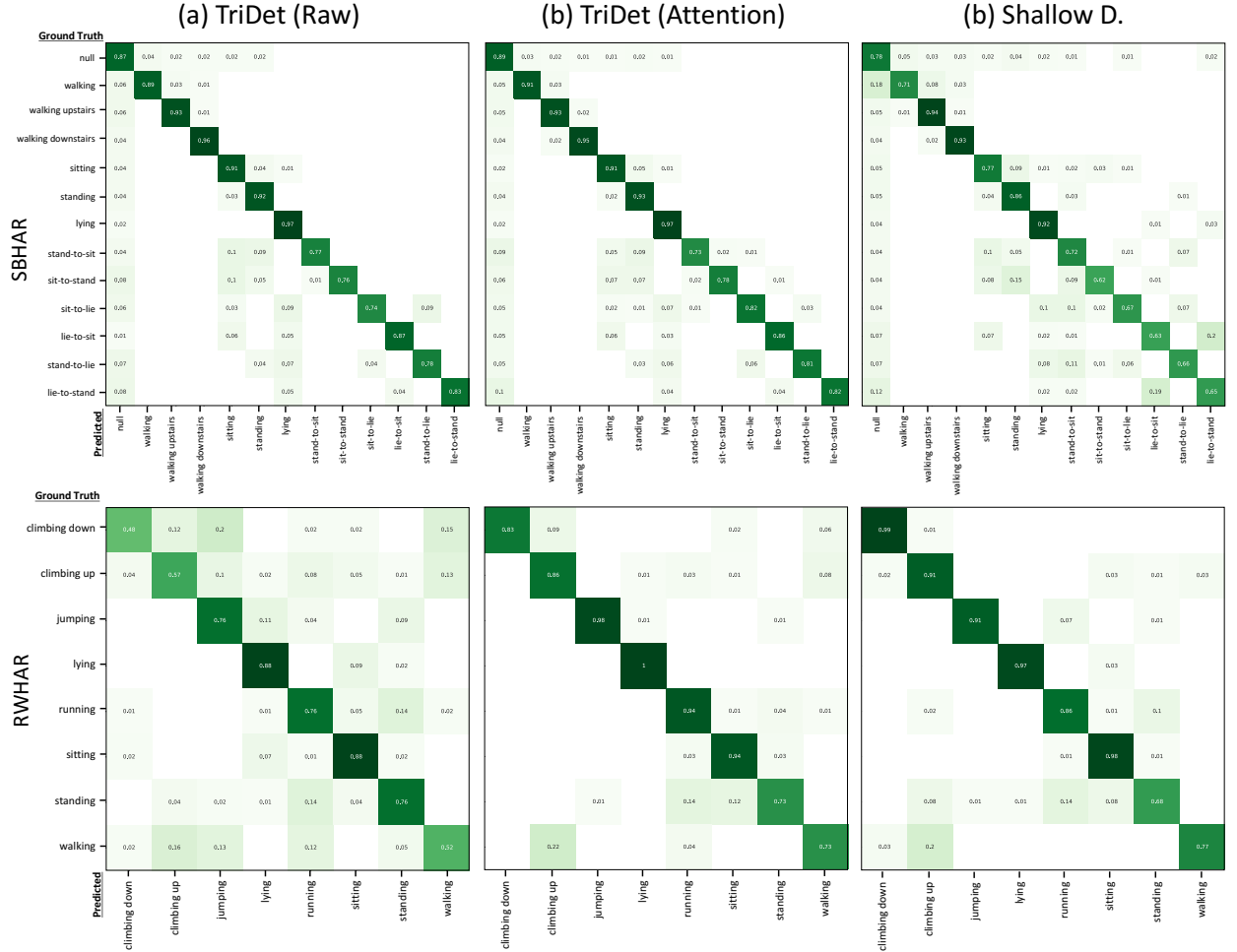


Figure 5.5.: Offline Activity Recognition: Confusion matrices of the (a) the best TAL architecture (TriDet) [151] and (b) inertial model (shallow DeepConvLSTM) being applied on the SBHAR [140] (top) and RWHAR dataset [164] (bottom) with a one second sliding window and 50% overlap. Note that confusions which are 0 are omitted.

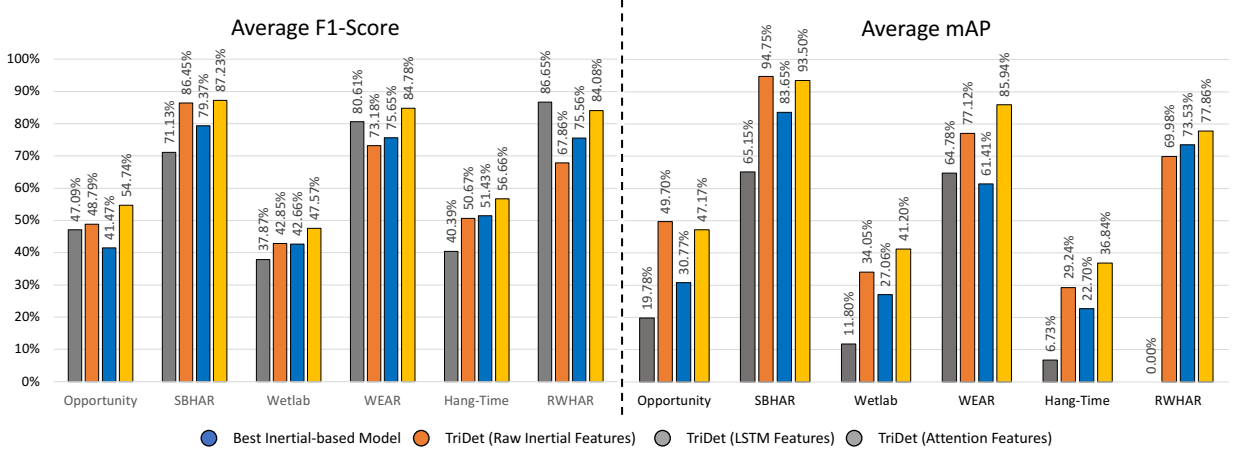


Figure 5.6.: *Offline Activity Recognition:* Average LOSO results of the two-stage TAL training of the TriDet model. We compare the LSTM and attention feature-based two-stage TAL training with the best inertial-based architecture and the single-stage TAL training on raw inertial data of all six benchmark datasets. One can see a clear improvement when using attention-based features across all datasets.

5.5.2. Online Activity Recognition

A major difference in how TAL models are intended to be used compared to inertial models is that they can only provide an offline prediction of previously recorded timeseries data. In previous experiments, we demonstrated that TAL models are capable of analyzing a participant’s data stream as a whole and can even deal with arbitrary lengths of timeseries data and activity durations. However, to investigate how capable TAL models are at analyzing only small snippets of incoming data streams, we modified the previously employed Leave-One-Subject-Out validation loop so that TAL models are tasked to predict each validation subjects’ data in a chunked manner. Specifically, given the data of a validation participant, we divided the data into equal-sized portions (e.g. 5 seconds worth of data) and had the TAL models, trained using unchunked data, predict each chunk individually. Table 5.3 compares the initial unchunked results of the tested TAL models (ActionFormer, TemporalMaxer and TriDet) on the six inertial benchmark datasets with results obtained when fragmenting each validation split into 1, 5, 30, or 60-second chunks. Across all datasets one can witness that with smaller-sized chunks classification and mAP results worsen across all algorithms and datasets. Overall, we can see that the smaller the chunk, the lower the overall confidence of the TAL models regarding each predicted segment. The lower confidence further leads to almost all segment predictions being removed even with a low score threshold such as 0.05, resulting in us not applying score thresholding in those cases, i.e., using all segments predicted by the respective TAL model. We assume that this drop in prediction confidence is likely caused by artificially splitting long segments of activities, contradicting what the models have seen during training, as datasets such as the Opportunity and Hang-Time dataset,

Table 5.3.: Average LOSO results obtained using the evaluated TAL architectures [204, 167, 151] employing different degrees of chunking during validation. During validation each subject is split into equal-sized chunks with each chunk being individually predicted by the trained TAL models. The table provides per-sample F1-Score (F1), average mAP applied at different tIoU thresholds (0.3:0.1:07) calculated on each chunk individually (c-mAP) and on the reconstructed validation stream (mAP). Results are averaged across three runs employing different random seeds. Results which are underlined are not score thresholded.

	Model	1 sec.			5 sec.			30 sec.			60 sec.			Unchunked	
		F1	c-mAP	mAP	F1	c-mAP	mAP	F1	c-mAP	mAP	F1	c-mAP	mAP	F1	mAP
Opp.	AF	<u>2.63</u>	<u>2.45</u>	<u>0.36</u>	37.73	18.10	25.51	49.96	41.74	45.84	50.50	45.45	47.84	51.93	51.24
	TM	<u>2.22</u>	<u>2.09</u>	<u>0.31</u>	38.00	19.08	25.99	45.32	39.24	42.83	45.27	42.15	44.22	44.74	46.31
	TD	<u>1.48</u>	<u>2.26</u>	<u>0.37</u>	9.96	<u>10.51</u>	<u>13.6</u>	<u>12.15</u>	<u>12.73</u>	<u>13.37</u>	<u>1.99</u>	<u>4.15</u>	<u>4.27</u>	48.79	49.70
SBHAR	AF	3.44	7.56	0.12	34.49	<u>26.20</u>	15.09	78.88	74.28	79.52	81.57	82.45	86.39	84.43	95.46
	TM	3.30	5.39	0.05	33.49	47.11	<u>23.87</u>	78.08	73.95	77.64	81.04	82.41	85.19	83.66	94.39
	TD	<u>2.13</u>	<u>10.18</u>	<u>0.13</u>	<u>17.69</u>	<u>29.31</u>	<u>8.05</u>	<u>31.70</u>	<u>52.01</u>	<u>49.38</u>	<u>44.76</u>	<u>72.94</u>	<u>72.64</u>	86.45	94.75
Wetlab	AF	5.13	<u>2.37</u>	0.02	<u>14.55</u>	<u>3.12</u>	<u>0.77</u>	35.35	25.65	20.15	35.98	30.46	28.06	40.71	33.53
	TM	<u>4.82</u>	<u>2.23</u>	0.03	<u>11.33</u>	<u>15.81</u>	<u>1.51</u>	34.37	25.87	20.16	35.35	30.24	26.90	37.09	35.72
	TD	<u>1.77</u>	<u>4.08</u>	0.03	<u>2.20</u>	<u>4.40</u>	<u>0.63</u>	<u>4.03</u>	<u>6.10</u>	<u>7.38</u>	<u>7.78</u>	<u>10.98</u>	<u>9.11</u>	42.85	34.05
WEAR	AF	5.70	3.47	0.00	15.54	10.44	0.06	53.57	40.02	18.72	69.21	56.89	52.11	72.43	73.80
	TM	5.30	2.65	0.00	38.37	53.39	0.19	58.29	45.09	18.94	67.91	56.60	47.89	69.52	69.18
	TD	<u>1.25</u>	<u>5.41</u>	0.00	<u>3.72</u>	<u>15.12</u>	<u>0.09</u>	<u>34.67</u>	<u>49.73</u>	<u>21.57</u>	<u>43.96</u>	<u>49.33</u>	<u>43.79</u>	73.18	77.12
Hang-T.	AF	<u>6.41</u>	<u>3.19</u>	<u>0.70</u>	42.41	17.18	14.67	49.32	26.47	26.82	50.69	27.64	27.65	51.23	29.26
	TM	<u>6.39</u>	<u>2.63</u>	<u>0.50</u>	41.13	17.54	15.17	48.15	25.72	25.38	48.67	26.63	26.33	47.17	27.86
	TD	19.52	0.00	3.56	19.33	3.67	3.48	26.02	12.42	13.01	35.72	18.80	19.15	50.67	29.24
RWHAR	AF	<u>8.47</u>	<u>28.67</u>	0.00	<u>17.00</u>	<u>51.96</u>	<u>0.00</u>	<u>50.15</u>	<u>76.19</u>	<u>1.37</u>	<u>62.40</u>	<u>77.58</u>	<u>7.91</u>	<u>61.24</u>	<u>65.40</u>
	TM	8.06	<u>17.13</u>	0.00	<u>31.35</u>	<u>48.89</u>	<u>0.00</u>	<u>52.71</u>	<u>60.47</u>	<u>1.31</u>	<u>58.47</u>	<u>61.86</u>	<u>7.61</u>	<u>60.59</u>	<u>50.60</u>
	TD	<u>5.88</u>	<u>25.68</u>	0.00	<u>9.07</u>	<u>40.11</u>	<u>0.00</u>	<u>39.78</u>	<u>47.78</u>	<u>0.94</u>	<u>48.65</u>	<u>57.51</u>	<u>7.39</u>	<u>67.86</u>	<u>69.98</u>

which contain mostly short-lasting segments, are less affected by the smaller chunks as e.g. the WEAR dataset which has almost exclusively segments lasting longer than 30 seconds. While the TriDet model is not capable of dealing with the chunked validation, most likely due to it using a different style of regression head, the ActionFormer and the TemporalMaxer are capable of maintaining predictive performance even for small-sized chunks of 5 seconds across almost all datasets. As expected, mAP scores are more impacted by the chunked prediction as chances are increased that long lasting segments are split due to the predicted segments not spanning across chunks. Nevertheless, we expect this effect to dampen if one would use an additional majority filtering as used in our inertial-based experiments, as this would eliminate potential intermediate activity switches.

Table 5.4 provides a comparison in terms of learnable parameters, size on disk, average batch training and inference times of the inertial-based models compared with the TAL models. One can see that the TAL models are significantly larger in size and number of learnable parameters. Though all architectures part of our analysis are capable of predicting a window of one second as well as the complete prediction stream of a participant within less than half a second, inertial-based models are faster than TAL models in both inference scenarios being on average faster by around a magnitude of 10. Taking into consideration that all TAL models fail to predict 1 second chunks and are significantly larger than inertial-based models, it becomes apparent, that TAL models are not suited for inference on edge-devices. It becomes apparent that with the models being trained to predict activity occurrences as a whole, the models require enough surrounding context in order to spot activities and correctly identify what they have seen during training. Nevertheless, as soon as the TAL models are given enough surrounding context (e.g., a prior context of 60 seconds), the models are capable to spot activities within said activity stream in a reasonable run time.

5.6. Discussion & Conclusion

This article demonstrated the applicability of vision-based, single-stage Temporal Action Localization for inertial-based Human Activity Recognition. We showed that three state-of-the-art TAL models [204, 167, 151] can be applied in a plain fashion to raw inertial data and achieve competitive results on six popular inertial HAR datasets [142, 140, 149, 164, 77, 19], outperforming in most cases popular models from the inertial community in an offline prediction scenario by a significant margin. Using a combination of an inertial network as a feature extractor and a TAL model showed to significantly enhance classification results for all TAL models – especially on datasets where classic inertial models had previously outperformed TAL models. Our two-stage experiments further suggest potential improvements that could be achieved by investigating improved methods for feature extraction using TAL models and

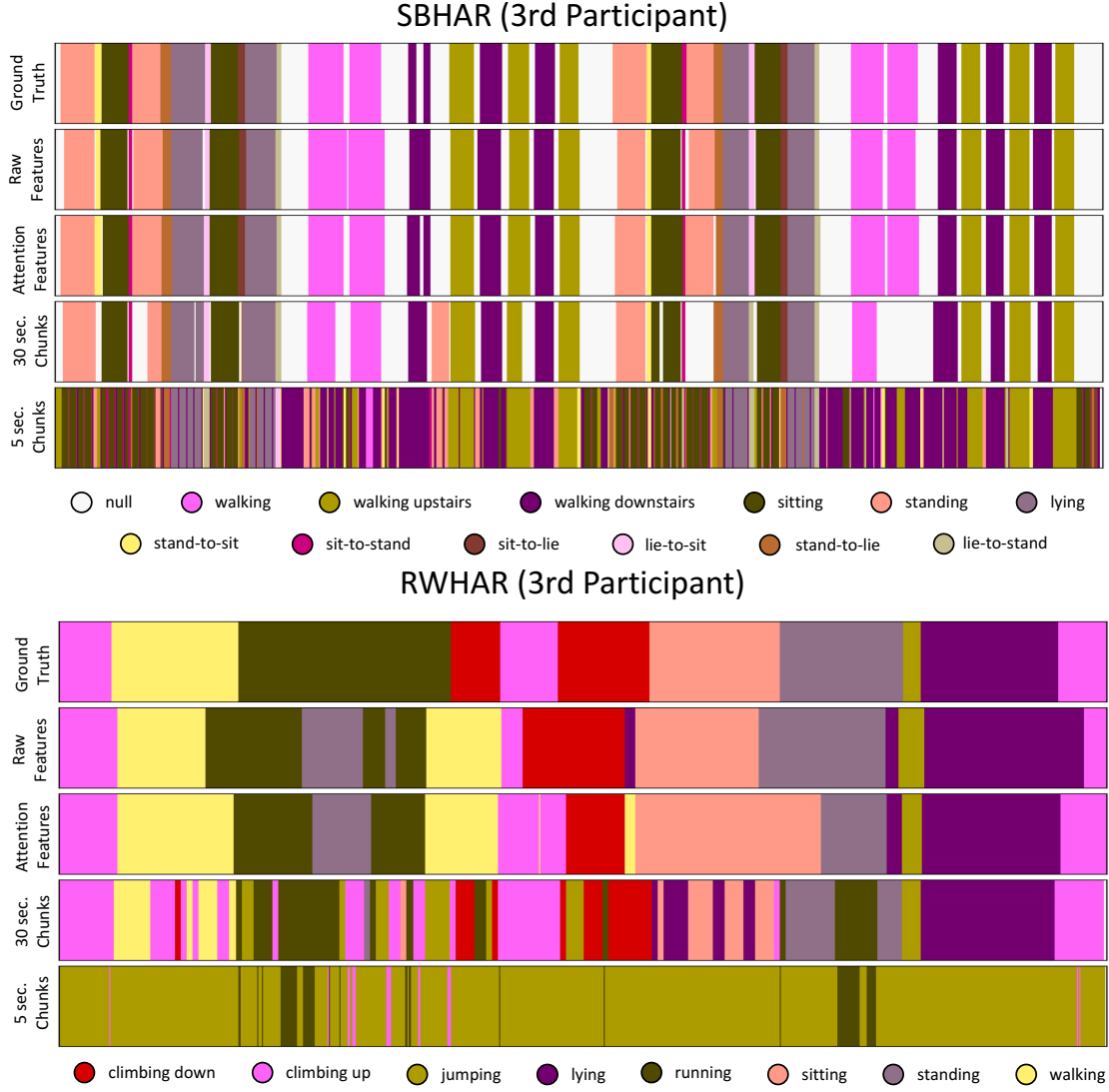


Figure 5.7.: Color-coded visualization of the effect of the *Offline* and *Online Activity Recognition* experiments using the ActionFormer model [204]. Results are compared with the ground truth of a sample participant as well as offline single-stage training for both the RWHAR [164] and SBHAR [140] dataset. One can see that training the TAL models using attention based features clearly improves results with an overall better recognition of all activities. Contrarily, once predicting a chunked output, performance can be somewhat maintained for larger chunks, yet drops near zero once applied on only 5-second long windows.

Table 5.4.: Comparison in terms of learnable parameters (in million (M)), size on disk (in MB), average batch training and inference times (in milliseconds) of the seven benchmark algorithms. Training and inference speeds are the average across the first 5 epochs of the first LOSO validation split of the algorithms being applied on the WEAR dataset. Benchmarking was performed on a single NVIDIA GeForce RTX 4090 with an AMD Ryzen 7800 X3D. We assess both inference speeds of the architectures being tasked to predict one 1-second window as well as one complete subject. ** 1 batch equals 100 1-second windows ** 1 batch equals 1 participant.

	Model	Total Params	Size	Avg. Time per Batch		
				Train	Test (1 sec.)	Test (1 participant)
Baseline	DeepConvLSTM	0.71 M	2.69MB	4ms*	2ms	27ms
	Shallow D.	0.57 M	2.19MB	3ms*	2ms	29ms
	A-and-D	0.57 M	2.19MB	31ms*	10ms	41ms
	TinyHAR	0.04 M	0.14MB	7ms*	3ms	28ms
	ActionFormer	27.02 M	103.37MB	131ms**	39ms	346ms
	TemporalMaxer	4.89 M	18.94MB	69ms**	25ms	380ms
	TriDet	13.75 M	52.73MB	127ms**	37ms	347ms

inertial data, e.g. via a fully-differentiable combined version of both type of architectures (see Figure 5.7).

A previously unexplored metric in inertial-based HAR, mAP, reveals that TAL models predict less fragmented timelines compared to inertial models and overall achieve larger degrees of overlap with ground truth segments. Furthermore, TAL models show to predict even non-periodic and complex activities more reliably than inertial architectures, providing consistent results across all types of classes across all datasets. Being one of the key challenges in HAR [26], the TAL architectures are further shown to be less affected by the unbalanced nature of HAR datasets due to a large NULL-class. Across the five benchmark datasets which offered a NULL-class, the TAL architectures showed to deliver the highest NULL-class accuracy. Additional experiments, which involved applying the TAL models on artificially created chunked sequences of data, showed that TAL models, though intended for the offline analysis of prerecorded activity timelines, have the capability of being applied in a near-online fashion (see Figure 5.7). Although the TAL models highlighted in this paper overall size and complexity would not allow them to be run on edge devices (yet), their reasonable performance and inference time on large enough chunks suggest that they could function e.g. as a server-side prediction model as seen in [12], which analyzes chunks of data in regular intervals.

The research community for inertial-based activity recognition has contributed methods to better model temporal relationships in the past years, yet most such architectures are limited to learning context within a fixed-sized sliding window. To this date, the length of

the sliding window remains a crucial parameter in inertial-based HAR, which may result in a significant performance drops in recognition performance when set incorrectly. Specifically, if dealing with both long and short lasting activities, a small sliding window might end up being too small to fully capture an activity, while a too large window size might cause sliding windows containing mixed activity types. With studies such as Pellatt and Roggen [130] and Guan and Plötz [61] contributing strategies to improve training, the TAL community offers an interesting new perspective to inertial-based HAR – alleviating the need for fixed size windows and making algorithms capable of dealing with activities of arbitrary length. Although models from these two independent communities share similarities, the TAL community offers many unexplored design choices and training concepts for a multitude of application scenarios, which we argue should be considered for investigation by the inertial-based HAR community.

6. DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activitiy Recognition

Title: DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activitiy Recognition

Authors: Marius Bock, Michael Moeller, Kristof Van Laerhoven

Venue: Not published, currently in review

Publication date: 27 May 2025

DOI: 10.48550/arXiv.2505.20894

Code: github.com/mariusbock/context_har

Combining ideas of both the previous work on TAL models for inertial HAR (see Chapter 5) as well as the later noted novel way of learning inter-window temporal patterns of the Shallow DeepConvLSTM (see Chapter 4), this work proposes a combined architecture, DeepConvContext, which introduces a novel multi-scale approach to inertial-based HAR. It herein revisits the Shallow DeepConvLSTM, as discussed in Chapter 4, highlighting the dimensional flip, which was introduced yet not discussed within the original publication, as means to perform inter-window temporal learning. The proposed architecture combines both inter- and intra-window learning outperforming both the original and Shallow DeepConvLSTM by a significant margin. Ablation experiments further highlight the superiority of LSTMs as compared to attention to learn temporal patterns in HAR. The preprint is as of submitting this dissertation still under review.

Contextualization

As discussed in the contextualization of Chapter 4, the dimensional flip present in the investigated architectures of the work discussed in Chapter 4 caused LSTMs part of the benchmarked models to be applied to the sequence of windows in a batch rather than across the convolutional features within a single window. While this contradicts the conventional definition of a batch, as it introduces temporal dependencies amongst batch elements, the results suggested that this form of inter-window temporal learning holds significant promise. As highlighted in the introduction of this dissertation (see Chapter 1) as well as within Chapter 5, the HAR community notoriously tends to rely on a sliding window classification

approach, in which models such as the DeepConvLSTM are trained to classify windows independently of each other. While enabling near-online inference, this strategy inherently limits a model’s ability to learn temporal dependencies across windows and remains a challenge that is widely acknowledged in the inertial-based HAR community [26, 73, 130].

The following chapter concludes efforts within this dissertation to address the the sliding window problem by introducing the DeepConvContext architecture, an inertial-based deep learning model that integrates both intra-window and inter-window context during classification. The model directly ties into design patterns of TAL models as described in Chapter 5, and extends the classic DeepConvLSTM [126] with a two-stage temporal modelling pipeline. Same as the original DeepConvLSTM, DeepConvContext first extracts per-window features using a combination of convolutions and in-window LSTMs, which are then additionally related using a second LSTM, learning inter-window temporal patterns within a batch. Benchmark experiments within this work demonstrate that DeepConvContext outperforms both the classic and Shallow DeepConvLSTM architectures across multiple HAR datasets. Ablation studies confirm initial observations from Chapter 4 that 1-layered LSTMs surpass 2-layered ones in performance, both for intra- and inter-window temporal learning. Additionally, when comparing attention-based methods to LSTMs, LSTMs consistently perform better, raising the hypothesis that their strong bias toward local temporal structure makes them particularly well-suited for HAR tasks, where long-range dependencies are often less critical. Overall, DeepConvContext represents a novel and effective application of temporal modelling in inertial-based HAR. The architecture effectively addresses the needs of the HAR community to extend temporal reasoning of models beyond a single sliding window. Unlike TAL models, DeepConvContext, retains efficiency through window-wise batching, rather than relying on full timeline classification. Despite current limitations of the data loader affecting the models ability to shuffle batches or enable real-time prediction, these issues can be addressed by developing a more flexible multi-scale data loader, which we judge remain subject for future research. Most importantly, the design choices proposed within the following work are transferable to other architectures, such as TinyHAR [210], making them a valuable contribution to ongoing developments in the HAR field.

Based on the finding of the dimensional flip within the Shallow DeepConvLSTM paper, Marius Bock suggested to investigate the combination of inter- and intra-window learning. A first architectural draft of the multi-scale classification approach was developed during discussion between Marius Bock, Michael Moeller and Kristof Van Laerhoven. Marius Bock conducted the literature review, implemented the architecture and conducted experiments. Ablation experiments comparing attention-based techniques with LSTMs were suggested by Michael Moeller. During write up of the final work, Michael Moeller and Kristof Van Laerhoven provided feedback and enhanced the clarity and readability of the final article.

Abstract

Despite recognized limitations in modeling long-range temporal dependencies, HAR has traditionally relied on a sliding window approach to segment labeled datasets. Deep learning models like the DeepConvLSTM typically classify each window independently, thereby restricting learnable temporal context to within-window information. To address this constraint, we propose DeepConvContext, a multi-scale time series classification framework for HAR. Drawing inspiration from the vision-based Temporal Action Localization community, DeepConvContext models both intra- and inter-window temporal patterns by processing sequences of time-ordered windows. Unlike recent HAR models that incorporate attention mechanisms, DeepConvContext relies solely on LSTMs – with ablation studies demonstrating the superior performance of LSTMs over attention-based variants for modeling inertial sensor data. Across six widely-used HAR benchmarks, DeepConvContext achieves an average 10% improvement in F1-score over the classic DeepConvLSTM, with gains of up to 21%. Code to reproduce our experiments is publicly available via www.github.com/mariusbock/context_har.

6.1. Introduction

Inertial-based HAR has, over the past two decades, notoriously relied on a sliding window-based classification approach [26]. Enabling near real-time prediction, this method divides continuous sensor data into overlapping windows, each of which is independently classified by a machine learning model such as the DeepConvLSTM [126]. While subsequent research has sought to enhance models’ capabilities through the integration of mechanisms like attention [210] to better model temporal patterns, these models remain fundamentally constrained by their dependence on sliding windows, restricting broader temporal understanding across windows [26].

In 2021, Bock et al. introduced the Shallow DeepConvLSTM [17], a more lightweight version of the DeepConvLSTM that removes the second LSTM layer. While this simplification led to improved performance across multiple HAR benchmark datasets, it also unintentionally introduced a novel form of inter-window learning. Specifically, the model applied the LSTM across windows within a batch, violating the traditional assumption that samples within a batch are independent. Although this design choice was not discussed in the original paper, suggesting it may have been unintentional, the reported results highlight the potential benefits of such cross-window temporal modeling. Similarly, the strategy of processing sequences of time-dependent sliding windows is well-established in a closely related field: single-stage TAL [204]. More recently, TAL models have been successfully applied to

inertial data and shown to outperform established HAR architectures such as DeepConvLSTM across various datasets [22]. As researchers continue to emphasize the importance of capturing both local and global temporal patterns in HAR [73], this paper introduces DeepConvContext, a novel inertial-based architecture inspired by TAL methodologies, which is specifically designed to model both short-term (intra-window) and long-term (inter-window) temporal dependencies across sequences of sliding windows.

Our contributions are three-fold:

1. We propose DeepConvContext, a novel architecture which employs a multi-scale recognition approach capable of capturing temporal dependencies both within and across a series of sliding windows.
2. An extensive evaluation on six widely-used HAR benchmark datasets, shows that DeepConvContext outperforms both the classical and Shallow DeepConvLSTM architectures, combining strengths of both architectures.
3. Ablation experiments show that LSTMs are capable of outperforming attention-based techniques by a significant margin, raising the question whether they remain a more suited choice for HAR.

6.2. Related Work

With the rise of deep learning, researchers have adapted convolutional neural networks (CNNs) for use on time series sensor data, eliminating the need for manual feature extraction that previously relied on domain expert knowledge. In 2016, Ordóñez and Roggen extended this approach by incorporating recurrent layers into inertial-based architectures, enabling them to capture temporal relationships between convolutional features [126]. In their original architecture, Ordóñez and Roggen employed a 2-layered LSTM as their recurrent method of choice. While many researchers have since improved upon the original DeepConvLSTM model [1, 210, 121, 191, 188], the architecture remains a viable and frequently used benchmark for inertial-based human activity recognition. Despite significant progress in improving methods for extracting temporal information, e.g. via including attention-mechanisms [1, 210, 121], architectures have historically relied on the sliding window approach, where each window of sensor data is processed independently to predict its label. As detailed by Bulling et al. [26], this approach, with its fixed window size, limits the model’s ability to capture inter-window temporal dependencies, making the choice of window length a critical factor in HAR performance, with an incorrectly chosen window size resulting in significant performance drops. Recent research has therefore emphasized the need for models that can

capture both local and global temporal patterns to overcome the limitations of the sliding window approach [73, 66].

In [130], Pellatt and Roggen propose *CausalBatch*, an improved training method for HAR, where windows within a batch are causally dependent on windows at the same position in neighboring batches. This allows the LSTM states to persist between batches, extending the model’s temporal horizon without increasing the window size. Similarly, Hiremath and Ploetz [73] explored the impact of larger context lengths on HAR performance, aggregating convolutional features using larger kernels before passing them through an LSTM. In [150], Shao et al. present an inter- and intra-window learning approach, which learns attention-based features both within and across windows, referred to as *frames* in their work. Compared to [130], our approach introduces a second LSTM specifically designed for learning inter-window context. Following the ideas proposed in [73], we aim to provide our model more flexibility by learning both local and global context separately at the cost of increased model complexity. This approach also aligns with trends in the TAL community, which separates local learning from global timeline reconstruction. In contrast to [150], our model relies on LSTMs, which, as shown in ablation experiments, outperform attention-based techniques.

6.3. Methodology

6.3.1. Architecture Overview

Figure 6.1 illustrates the data flow of the proposed DeepConvContext architecture. This architecture follows a multi-scale approach, incorporating methods for extracting temporal features from inertial sensor time series, inspired by both the original [126] and Shallow DeepConvLSTM [17]. Given a multivariate time series from a set of inertial sensor axes, DeepConvContext begins by segmenting the sequence into equal-sized patches using a sliding-window approach. Features are extracted from each patch through multiple convolutional layers, followed by an LSTM, which mirrors the feature extraction process of the original DeepConvLSTM. This stage is designed to capture discriminative temporal dependencies within each sliding window. Unlike the original DeepConvLSTM, which performs classification based on the final element of each patch, DeepConvContext introduces an additional stage that captures dependencies across patches. This inter-patch learning mechanism is inspired by the design of the Shallow DeepConvLSTM. To achieve this, each patch is reduced to a one-dimensional feature vector of size $1 \times LSTM_h$, where $LSTM_h$ represents the number of hidden units in the last layer of the intra-patch LSTM. These patch representations are then treated as a new sequential input to a second LSTM, which models temporal dependencies across patches. The output of this second LSTM is passed through a dropout layer and

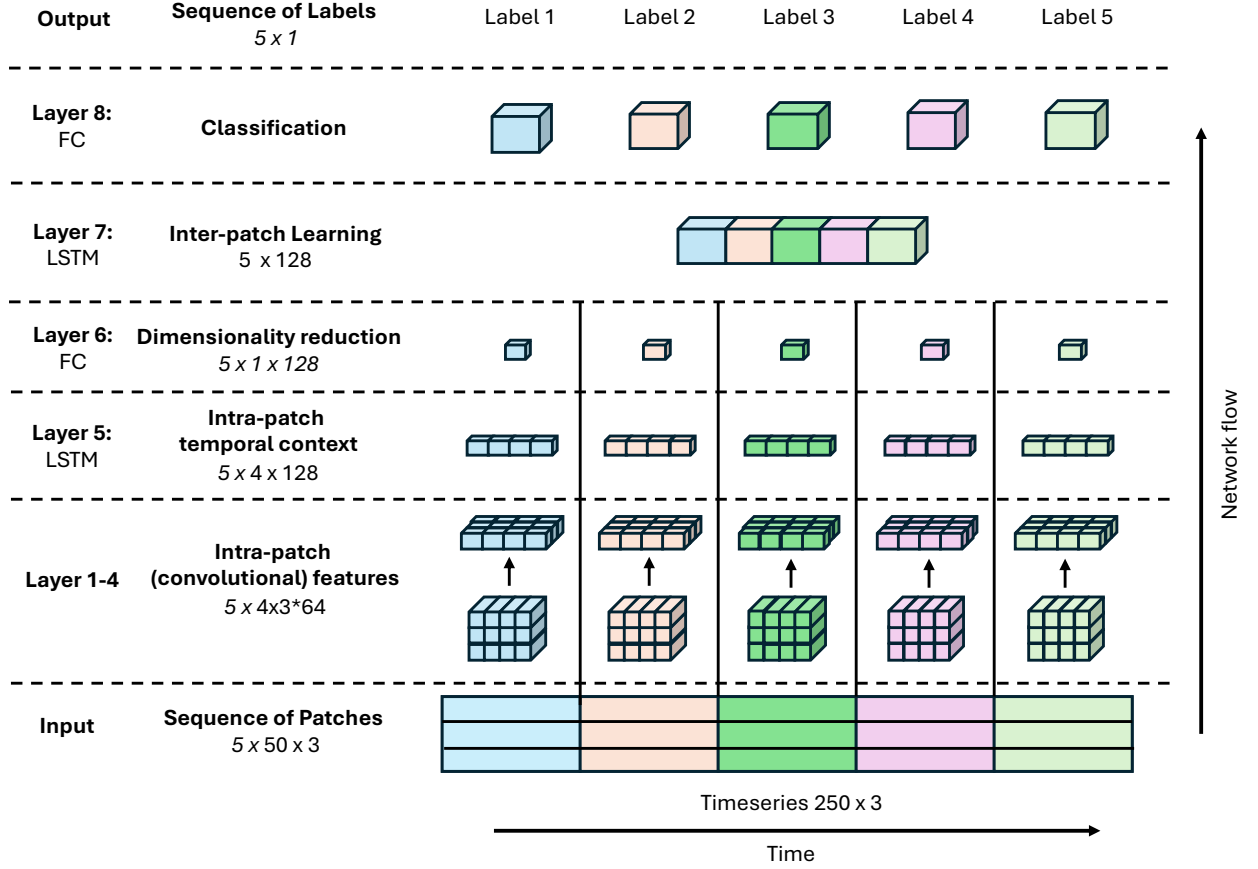


Figure 6.1.: Overview of the proposed *DeepConvContext* architecture. The architecture follows a multi-scale approach in which an input timeseries is segmented into equal-sized patches. These patches are individually processed by a *DeepConvLSTM*-like feature extraction, i.e. a combination of multiple convolution and a LSTM. The resulting intra-patch temporal context vectors are dimensionally reduced to a 1-dimensional feature vector. The sequence of feature vectors of all patches are then passed to a second LSTM, to learn inter-patch temporal features. Resulting patch-wise feature vectors are then classified and a sequence of patch-wise activity labels is returned.

a fully connected classification layer. Because each patch is processed by both intra- and inter-patch LSTMs, it retains information about its internal temporal structure, as well as its temporal relationship with earlier patches in the sequence.

6.3.2. Datasets

We base our experiments on six widely used HAR datasets. These include the Wetlab [149], WEAR [20], SBHAR [140], RWHAR [164], Opportunity [142], and Hang-Time datasets [77]. This selection provides a diverse set of prediction scenarios, each posing distinct challenges for activity recognition models to address. Among these challenges are variations in the number of participants and activity classes, the presence of a NULL-class [20, 149, 77, 142], and the inclusion of complex, short-duration, transitional, and locomotion activities [20, 77,

142, 149, 140, 164]. Additionally, the datasets vary in sensor configurations, with several collected in multi-sensor environments [20, 142, 164].

6.3.3. Architecture Variants

By default, LSTMs operate in an unidirectional manner, processing input sequences from past to present. Within the DeepConvContext architecture, this means that patches, i.e. windows, are fed sequentially into the model, with the first patch corresponding to the earliest time point and each subsequent patch representing the next point in time. This design limits the LSTM to only past and present information, preventing it from incorporating any future context into its predictions. However, certain activities, such as *sit-to-stand* present in the SBHAR dataset [140], can benefit from access to future context during prediction. To address this, we also evaluate a variant of the DeepConvContext that uses a bidirectional LSTM to model inter-patch temporal dependencies. In this variant, the second LSTM within the DeepConvContext module (depicted in Figure 6.1) processes each batch sequence in both forward and backward temporal directions. For every patch, it outputs two vectors of size $1 \times LSTM_h$, where $LSTM_h$ is the number of hidden units in the LSTM. These vectors are then concatenated to form a combined representation, which is passed, just like in the unidirectional case, into a fully connected layer that serves as the classifier.

LSTMs, though effective at modeling local temporal dependencies, often struggle to learn long-range relationships when the temporal distance between relevant sequence elements increases [11, 127]. In contrast, self-attention mechanisms can capture dependencies between all elements of a sequence, regardless of their relative position in time. While this capability comes with increased computational overhead, it has led to notable performance gains in inertial sensor-based architectures such as Attend-and-Discriminate [1] and TinyHAR [210]. In line with these developments, we further evaluate two additional variants of the DeepConvContext architecture: one using multi-head self-attention and another incorporating a transformer module, as described in [93]. To maintain consistency with the LSTM-based setups, we implement unidirectional versions of both attention-based mechanisms, specifically, causal attention and causal transformers, to ensure that each model only learns from past temporal information.

6.3.4. Training

For each experiment, we perform LOSO cross-validation, where each participant in the dataset is used as the validation set once, while all other participants are used for training. We report two key metrics for each experiment: the class-averaged macro F1-score and the mAP, both averaged across all validation splits (i.e., participants). The mAP, which

evaluates the overlap between predicted activity segments and ground truth annotations, is computed across five different tIoU thresholds. To calculate both mAP and F1-score, we first convert the windowed predictions of each model back to per-sample predictions by unwinding them. This ensures that both metrics are computed on the same temporal resolution. The final reported values per dataset are the averages of the tIoU-averaged mAP and class-averaged F1-score across all validation subjects. All model architectures are trained using a weighted cross-entropy loss and the Adam optimizer, with a learning rate of $1e^{-4}$, weight decay of $1e^{-6}$. Each model is trained for 30 epochs, using a learning rate schedule that multiplies the learning rate by a factor of 0.9 every 10 epochs. A training batch size of 100 is used for all experiments. For evaluation, the original DeepConvLSTM is tested using a batch size of 1, while models that rely on inter-window features, namely the Shallow DeepConvLSTM and DeepConvContext, are evaluated with a test batch size equal to the training batch size (100), to maintain the temporal structure required for inter-window learning. All architectures evaluated in this benchmark use four convolutional layers with 64 filters, each of size 9×1 , except for the Opportunity dataset, where a smaller filter size of 5×1 is used due to the lower sampling rate of the input signal. LSTM layers are configured with a hidden size of 128 and no dropout between layers. In ablation experiments where LSTMs are replaced with transformers or multi-head self-attention mechanisms, we use four attention heads with no dropout. The transformer-based variants use three layers. Same as Ordóñez and Roggen [126], embeddings get passed through a dropout layer with a probability of 0.5 before being passed to the classifier.

6.4. Results

6.4.1. Benchmark analysis

In Figure 6.2, we present an overview of the results for the classic DeepConvLSTM [126], the Shallow DeepConvLSTM [17], and the proposed DeepConvContext model, all using a single LSTM layer per module. On average, the multi-scale DeepConvContext achieves the highest prediction performance, with an average F1-score of 59.04%, compared to 49.09% for the intra-window DeepConvLSTM and 54.01% for the inter-window Shallow DeepConvLSTM, across the six evaluated datasets. When comparing the intra-and inter-window models, we observe that the inter-window DeepConvLSTM struggles to produce coherent activity segments, reflected by its average mAP being close to zero across all datasets. In contrast, both the Shallow DeepConvLSTM and DeepConvContext leverage temporal relationships across windows and thus produce fewer rapid switches between activities, achieving significantly higher average mAP values of 16.79 and 18.23, respectively. It is worth noting that the

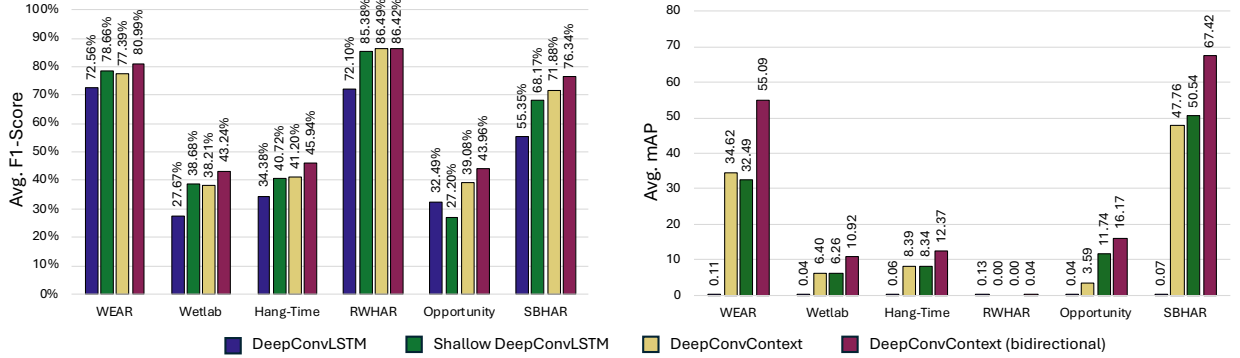


Figure 6.2.: Average F1-score and mAP results of the DeepConvLSTM [126], Shallow DeepConvLSTM [17] and proposed DeepConvContext being applied to the WEAR [20], Wetlab [149], Hang-Time [77], RWHAR [164], Opportunity [142] and SBHAR [140]. The DeepConvContext is additionally evaluated using bidirectional LSTM to perform inter-window learning. Results are the class- and participant-averaged scores averaged across three runs using different random seeds. One can see that the DeepConvContext combines strengths of both architectures and improves upon results across all datasets, with the bidirectional version of the architecture providing the highest F1-score and mAP.

RWHAR dataset does not include a NULL class and contains only a single segment per activity per participant. This makes mAP evaluation particularly sensitive, since the metric is biased toward zero unless a model predicts an activity nearly perfectly across its full duration in the participant’s data stream.

The Shallow DeepConvLSTM outperforms the classical DeepConvLSTM on all datasets except Opportunity. This highlights the importance of inter-window context for accurate activity recognition, as the Shallow DeepConvLSTM does so by only relating convolutional features of the final sequence element from each window in a batch. It is capable of recognizing both periodic activities such as *sitting* (found in datasets like SBHAR and RWHAR) and non-periodic or transitional activities such as *sit-to-lie* in SBHAR. However, as shown in the confusion matrices in Figure 6.3, the DeepConvLSTM still outperforms the Shallow DeepConvLSTM for strictly periodic activities. This is likely because such activities are often characterized by short, repeating patterns that can be effectively captured by convolutional filters and do not require modeling of long-range temporal dependencies [16]. Understanding this, the DeepConvContext effectively combines the strengths of both intra- and inter-window learning, maintaining high predictive performance on all activity types. Finally, Figures 6.2 and 6.3 also show the results for a variant of DeepConvContext that uses a bidirectional LSTM for inter-window learning. By incorporating information from future windows, this variant improves the accuracy of predictions for individual windows and leads to a substantial increase in overall mAP.

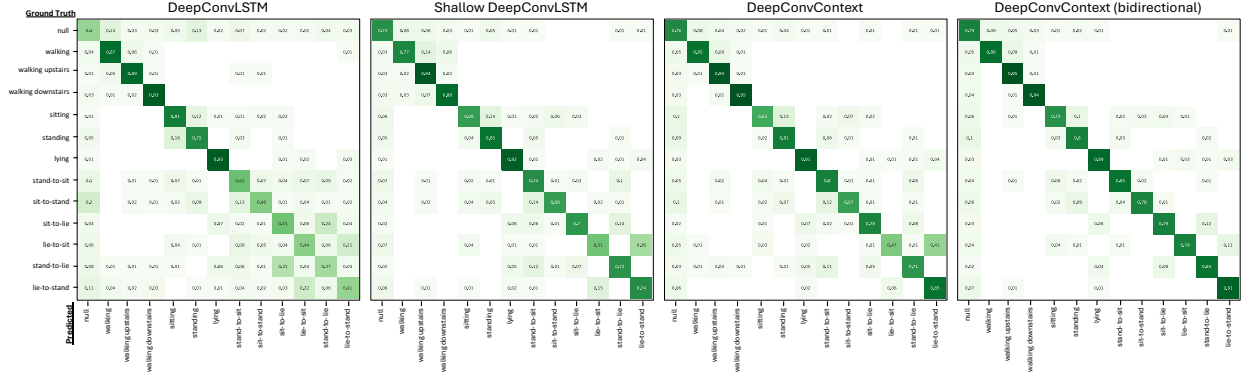


Figure 6.3.: Per-class confusion matrices of the DeepConvLSTM, Shallow DeepConvLSTM and DeepConvContext being applied to the SBHAR dataset using LOZO cross-validation. The DeepConvContext is further applied using bidirectional LSTM as described in Chapter 6.3.3. One can see that the DeepConvContext improves upon both variants of the DeepConvLSTM, with the bidirectional variant producing the overall highest prediction results. Especially transition classes such as *sit-to-stand* are more reliably detected.

6.4.2. Ablation experiments

Revisiting shallow LSTMs When introducing the Shallow DeepConvLSTM in [17], Bock et al. demonstrated that shallow LSTMs might be a better option for inertial-based activity recognition, challenging the widely held belief based on the earlier findings of Karpathy et al. [89]. However, since the original Shallow DeepConvLSTM paper unintentionally focussed on the use of LSTMs for learning inter-window features, their analysis does not provide evidence to conclude that shallow LSTMs are superior for learning intra-window dependencies. To revisit this claim, we conducted a comparative evaluation using the original DeepConvLSTM, the Shallow DeepConvLSTM, and our proposed DeepConvContext model. Each architecture was tested with both single-layer and two-layer LSTM configurations. As shown in Table 6.1, the results indicate that single-layer LSTMs tend to outperform their two-layer counterparts across all datasets, with the exception of the Opportunity dataset [142]. Moreover, the performance gains from using shallower LSTMs are more significant when applied to inter-window learning. We attribute this due to the longer inter-window sequences likely containing a richer set of learnable patterns than intra-window sequences, which may make the limitations of deeper LSTMs more evident and highlight the advantages of shallow 1-layered variants.

LSTMs vs. Attention With the rise in popularity of attention-based mechanisms, architectures such as those proposed in [1] have demonstrated their applicability to HAR. However, as previous studies have primarily focused on architectures designed to learn intra-window features, we conducted further experiments to investigate whether multi-head attention [177]

Table 6.1.: Average F1-score results of the DeepConvLSTM [126] (DC-LSTM), Shallow DeepConvLSTM [17] (Shallow D.) and proposed DeepConvContext using either 1-layered (1-L) or 2-layered LSTMs (2-L) within the architectures. We report results for the WEAR [20], Wetlab [149], Hang-Time [77], RWHAR [164], Opportunity [142] and SBHAR [140]. Results are the class- and participant-averaged scores averaged across three runs using different random seeds. One can see that one-layered LSTMs outperform 2-layered LSTMs across all architectures and datasets, except the Opportunity dataset. Best results per dataset are underlined.

D	DC-LSTM		Shallow D.		DeepConvContext	
	1-L	2-L	1-L	2-L	1-L	2-L
[20]	72.56%	70.04%	<u>78.66%</u>	74.27%	77.39%	75.12%
[149]	27.67%	26.26%	<u>38.68%</u>	33.64%	38.21%	33.61%
[77]	34.38%	33.76%	40.72%	38.24%	<u>41.20%</u>	39.80%
[164]	72.10%	70.90%	85.38%	83.36%	<u>86.49%</u>	84.42%
[142]	32.49%	37.66%	27.20%	31.03%	<u>39.08%</u>	33.39%
[140]	55.35%	54.37%	68.17%	63.50%	<u>71.88%</u>	61.03%
<i>Avg</i>	<u>49.09%</u>	<u>48.83%</u>	<u>56.47%</u>	<u>54.01%</u>	<u>59.04%</u>	<u>54.56%</u>

and ViT-style Transformers [93] could serve as more effective alternatives to LSTMs for modeling temporal dependencies between windows. To explore this question, and as described in Section 6.3.3, we replaced the second LSTM in the DeepConvContext architecture with either a multi-head attention block or a Transformer module. As shown in Table 6.2, our results indicate that LSTMs outperform attention-based mechanisms across the selected benchmark datasets. In particular, for the Wetlab and Hang-Time datasets, LSTMs achieve an average F1-score that is approximately 10% higher than that of the Transformer-based variants. While attention mechanisms allow features to interact across the entire input sequence, they typically exhibit a reduced sensitivity to the natural order of events in time and rely on positional encodings to provide temporal structure. Although we included such encodings in our models, the relatively lower performance of the attention-based variants suggests that LSTMs may be more effective at capturing temporal relationships among windows. Since many human activities do not require excessively long temporal contexts for accurate classification, LSTMs may be better suited for such HAR tasks due to their strength in modeling localized temporal patterns. This assumption is further supported by the improved performance of attention-based DeepConvContext variants on the RWHAR and SBHAR datasets. In these cases, the long and sequential nature of activity segments allows attention layers to learn extended temporal dependencies more effectively. These dependencies enable the model to reproduce the long, coherent activity sequences present in the data patterns, which LSTMs may struggle to model as efficiently.

Table 6.2.: Comparison of average F1-score results of the DeepConvContext using either an LSTM, attention or a Transformer to learn inter-window context. We assess both unidirectional and bidirectional versions of the three techniques. We report results for the WEAR [20], Wetlab [149], Hang-Time [77], RWHAR [164], Opportunity [142] and SBHAR [140]. Results are the class- and participant-averaged scores averaged across three runs using different random seeds. One can see that on average LSTMs outperform both attention and Transformers. Best unidirectional and bidirectional results per dataset are underlined.

D	Unidirectional			Bidirectional		
	LSTM	Att.	Tran.	LSTM	Att.	Tran.
[20]	<u>77.39%</u>	76.21%	76.15%	<u>80.99%</u>	80.25%	75.34%
[149]	<u>38.21%</u>	31.64%	28.81%	<u>43.24%</u>	37.45%	28.03%
[77]	<u>41.20%</u>	38.44%	37.85%	<u>45.94%</u>	42.45%	36.62%
[164]	86.49%	<u>86.59%</u>	63.60%	86.42%	<u>89.71%</u>	62.14%
[142]	<u>39.08%</u>	36.68%	38.96%	<u>43.96%</u>	32.36%	33.79%
[140]	71.88%	67.43%	<u>75.65%</u>	<u>76.34%</u>	71.25%	76.15%
<i>Avg</i>	<u>59.04%</u>	56.17%	53.50%	<u>62.82%</u>	58.91%	52.01%

6.5. Limitations

The DeepConvContext architecture presents a novel approach to processing inertial sensor data for activity recognition. Although not specifically tested within this work, its architectural design choices can be readily extended to other inertial-based models such as TinyHAR [210] and Attend-and-Discriminate [1]. While our analysis evaluates only modified versions of the original DeepConvLSTM, we expect the observed performance improvements to generalize to other architectures, as they follow a common design pattern of combining convolutional feature extraction with some form of temporal modeling. Compared to CausalBatch [130] as well as related work [73], our architecture introduces some additional computational complexity, as shown in Table 6.3. However, by incorporating a projection layer between the two LSTMs, the overall increase in GPU memory usage and floating point operations remains modest, being approximately 3.5% and 10%, respectively. Furthermore, while Table 6.3 indicates that the attention-based version of DeepConvContext uses fewer learnable parameters, LSTMs are slightly more computational efficient. In our experiments, we did not modify the data loading strategy of the evaluated HAR datasets, apart from disabling batch shuffling to maintain temporal consistency among batch elements. This consistency is essential, and the same principle must be applied during testing. For instance, using a test batch size of 1 would prevent the model from accessing any contextual information across patches, thereby negating the benefits of inter-window learning. In real-world applications, this constraint means that DeepConvContext cannot be directly applied in an online setting. Instead, it requires data to be collected for a duration of $b \times (w - o)$ seconds, where b is the training batch size, w is the sliding window length in seconds, and o is the overlap in

Table 6.3.: Model complexity of the original DeepConvLSTM (DC-LSTM), Shallow DeepConvLSTM (Shallow D.) and architecture variants of the DeepConvContext (DCC) in terms of number of learnable parameters (ρ), GigaFLOPS (GFLOPS) and GPU memory consumption (Memory). All architectures apply 1-layered LSTMs, if applicable. Attention-based architectures use 4 attention heads as well as 3 Transformer layers. One can see that the DeepConvContext has around 2.5 times more learnable parameters, yet only introduces a 3.5% and 10% increase in GFLOPS and memory consumption.

Architecture	ρ	MFLOPS	Memory
DC-LSTM	277,062	1,739.98	114.42MB
Shallow D.	277,062	1,739.98	114.42MB
DCC (LSTM)	704,198	1,798.96	124.21MB
DCC (Bi-LSTM)	837,062	1,799.19	128.09MB
DCC (Att.)	638,150	1,817.22	123.20MB
DCC (Tran.)	1,961,158	2,074.42	143.39MB

Table 6.4.: Average F1-score and mAP results of the DeepConvContext using varying train and test batch sizes (25, 50 and 200). We report results for the WEAR [20], Wetlab [149], Hang-Time [77], RWHAR [164], Opportunity [142] and SBHAR [140]. Results are the class- and participant-averaged scores averaged across three runs using different random seeds. One can see that on average LSTMs outperform both attention and Transformers.

D	25		50		200	
	F1	mAP	F1	mAP	F1	mAP
[20]	74.25%	22.81	76.44%	26.02	78.37%	37.44
[149]	30.57%	1.83	35.35%	4.01	38.32%	6.78
[77]	40.00%	7.66	40.44%	8.32	41.73%	8.72
[164]	50.23%	0.02	84.09%	0.06	85.68%	0.00
[142]	31.04%	6.26	34.68%	7.81	39.28%	10.84
[140]	66.46%	38.08	68.25%	44.48	74.64%	55.66
<i>Avg</i>	<i>48.76</i>	<i>12.78</i>	<i>56.54</i>	<i>15.12</i>	<i>59.67</i>	<i>19.91</i>

seconds. This makes the context length a hyperparameter which needs to be considered during training of a network as it influences recognition of activities (see Table 6.4). Future research could focus on designing an optimized data loader that maintains a running memory of previously seen windows. This would enable the architecture to function according to a first-in, first-out (FIFO) principle, where features are computed only for the newly recorded window and then related to past context. Such a strategy would allow for an efficient and online deployment of the DeepConvContext architecture. Alternatively, one could extend the data loading process by extending the sliding window input to a context dimension of previously recorded data. This would allow for shuffling batch elements, possibly resulting in improved results as architectures would be optimised on alternating contexts and activities.

6.6. Discussion & Conclusions

In this paper we introduced DeepConvContext, a novel architecture for inertial-based activity recognition. Inspired by advancements in the vision-based Temporal Action Localization community, the architecture adopts a multi-scale approach that learns both intra- and inter-window temporal dependencies to predict the label of a given sliding window. Our analysis revealed that although this was not discussed in the original publication, the Shallow DeepConvLSTM [17] introduced a unique method for learning inter-window temporal features by implicitly relating windows within a batch. While this approach challenges the conventional definition of a batch, our results highlight the importance of inter-window learning. Benchmark experiments showed that inter-window learning consistently outperforms intra-window learning, as used in the original DeepConvLSTM [126], across a range of Human Activity Recognition datasets. DeepConvContext builds on the strengths of both prior architectures by integrating both intra- and inter-window learning into a single unified model. Our benchmark analysis showed that on average, our architecture outperforms both the original and Shallow DeepConvLSTM, with gains up to 21% in F1-score. Further ablation experiments, which involved testing architectural variants of DeepConvContext, demonstrated that average mAP can be significantly improved by making the second LSTM, which is responsible for inter-window learning, bidirectional. Additional experiments showed that LSTMs are more effective than attention-based mechanisms, raising the question of whether their strong focus on local temporal structures makes them better suited for recognizing human activities, which often do not require long range temporal connections.

With DeepConvContext, we present the HAR community with a novel method for processing windowed data and predicting activity labels. We directly address the well-known sliding window problem [26] and align with the community’s ongoing efforts to overcome it [130, 73]. Although our current implementation does not yet incorporate an optimized multi-scale batch data loader, we consider this a promising direction for future research and expect it to further enhance the performance of our architecture. With our suggested changes being easily extendable to other architectures, we hope that our proposed multi-scale approach will become an essential component within future works.

7. Conclusion

While Deep Learning has largely automated feature extraction from inertial data, reducing the dependence on expert-crafted features, it has inadvertently introduced new challenges to the activity recognition community. This dissertation addressed two of these core challenges brought about by the adoption of Deep Learning in inertial-based HAR: the *labeling bottleneck* and the *sliding window problem*.

Chapters 2 and 3 examined the *labeling bottleneck*, a persistent challenge in inertial-based HAR arising from the substantial manual effort required for data annotation and the inherent difficulty in retrospectively interpreting inertial sensor data. This limitation has led to comparatively small benchmark datasets, impeding the development and evaluation of data-intensive Deep Learning models. Though videos remain a favorable choice to overcome the lack of interpretability of inertial sensors, datasets that provide both modalities while employing a sensor layout driven by real-world application scenarios remain limited. Chapter 2 thus presented a novel, multimodal benchmark dataset, called WEAR, specifically designed to address the limitations of existing benchmark datasets. The dataset employs a standardized and easily replicable wearable sensor configuration, paired with minimal requirements for participants and recording conditions, thereby ensuring ease of extensibility. Recorded at more than ten outdoor locations over the span of multiple months, the resulting dataset offers diverse visual contexts with varying weather and environmental conditions. With the WEAR dataset providing a complementary sensor layout of video and inertial data, Chapter 3 directly aimed to address the labeling bottleneck by proposing a weak-annotation pipeline that exploits the discriminative feature extraction capabilities of vision foundation models to cluster time-synchronized video recordings of HAR experiments. By showing annotators only representative clips instead of requiring them to annotate full videos, the pipeline significantly reduces the annotation burden associated with HAR data, using clustering information to propagate labels to similar clips. Further empirical results demonstrated that the resulting weakly annotated datasets are effective for training deep learning classifiers.

As prior work conducted as part of this dissertation showed the capabilities of novice annotators for wearable data annotation [76], the proposed annotation tool paired with a wearable sensor setup such as that of the WEAR dataset, could enable participants to

easily collect their own in-the-wild data, while being asked for a more reasonable annotation burden that could improve long-term study retention. Future extensions of these works thus might involve integrating energy-efficient video capture technologies, such as those employed in datasets like CAPTURE-24 [172], or wearable platforms like Project Aria [49] and Ego-Exo4D [60]. However, such efforts must ensure that inertial sensor placement is optimized to preserve the modality’s discriminative potential, an aspect that is often underappreciated in vision-centric data collection protocols. Furthermore, as the annotation pipeline proposed in Chapter 3 only uses visual data to cluster experimental videos, other modalities, especially the recorded inertial data, could be included to improve the performance of the clustering, such as shown in [53].

Chapter 4, 5 and 6 addressed the *sliding window problem*, specifically focusing on the capabilities of inertial models in capturing temporal dependencies. Chapter 4 revisited the widely adopted DeepConvLSTM model [126], demonstrating that a simplified variant employing a single LSTM layer can yield improved performance across multiple datasets, thereby challenging prevailing assumptions about the necessity of deep recurrent stacks for effective sequence modeling [89]. With the HAR community notoriously tending to rely on a sliding window classification approach, Chapter 5 challenged this approach by demonstrating the application of vision-based TAL models for segment-level classification of inertial sequences. These models consistently outperformed traditional inertial-based architectures, producing less fragmented activity timelines, while being able to handle short and long lasting activities simultaneously. Building on this, Chapter 6 introduced DeepConvContext, a novel architecture extending the classic DeepConvLSTM that enable inter-window temporal learning by incorporating a second LSTM across batch dimensions. While this approach deviates from standard batching conventions, our design choice was based on a later noted dimensional flip within the architectures investigated in Chapter 4, which caused the LSTMs to be applied in a similar fashion. By combining inter- and intra-window temporal relations, the architecture effectively captures higher-level temporal dependencies and achieves superior performance compared to the classic DeepConvLSTM. Ablation experiments using variants of the architecture further showed LSTMs to be more effective than attention-based mechanisms, raising the question of whether their strong focus on local temporal structures makes them better suited for recognizing human activities, which often do not require long range temporal connections.

The TAL community offers a compelling new perspective for inertial-based HAR. Its emphasis on reconstructing complete activity timelines, rather than producing isolated window-based predictions, reduces the sensitivity of model performance to hyperparameters chosen during segmentation. While TAL models have demonstrated notable flexibility with respect to input sequence length and feature types, they are still ill-suited for deployment on

resource-constrained edge devices. Although Chapter 6 partially integrates design patterns from TAL into the inertial-based DeepConvLSTM framework, further research is required, particularly in the development of optimized data loaders capable of supporting multi-scale temporal modeling and efficient ways to handle introduced complexity within these models. Nonetheless, Chapters 4, 5, and 6 propose novel architectural design choices that are hoped to be adopted by the community and serve as a valuable contribution to ongoing methodological advancements. While other domains have rapidly transitioned to new techniques such as attention-based mechanisms, the demonstrated superiority of LSTMs in Chapter 6 suggests that such transitions within the HAR community should be approached with caution – considering the specific temporal structure inherent to human activities.

To conclude, this dissertation addressed two substantial problems faced within inertial-based HAR through the application of computer vision methods. The demonstrated use cases show how recent advancements in video-based activity recognition can help overcome limitations inherent to inertial sensing. While each approach exhibits certain constraints, these works offer novel perspectives on long-standing issues and introduce methodologies that, if adopted, could inspire further research and innovation within the inertial HAR community. In more general terms, this dissertation promotes increased methodological exchange and collaboration across research communities and sensing modalities. Each modality, with its own set of strengths and limitations, has led each community to develop distinct strategies to address them. While these approaches may initially appear unrelated, they often share inherent similarities. As demonstrated in this work, identifying and adapting such shared elements can offer valuable solutions to persistent challenges – not only within inertial-based HAR, but across the broader landscape of activity recognition.

Bibliography

- [1] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezaatofghi, and Damith C. Ranasinghe. 2021. Attend and Discriminate: Beyond the State-Of-The-Art for Human Activity Recognition Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021). <https://doi.org/10.1145/3448083>
- [2] Alireza Abedin, Farbod Motlagh, Qinfeng Shi, Hamid Rezaatofghi, and D. Ranasinghe. 2020. Towards Deep Clustering of Human Activities from Wearables. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3410531.3414312>
- [3] Rebecca Adaimi and Edison Thomaz. 2019. Leveraging Active Learning and Conditional Mutual Information to Minimize Data Annotation in Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019). <https://doi.org/10.1145/3351228>
- [4] Abrar Ahmed, Harish Haresamudram, and Thomas Ploetz. 2022. Clustering of Human Activities from Wearables by Adopting Nearest Neighbors. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3544794.3558477>
- [5] Bandar Almaslukh, Jalal AlMuhtadi, and Abdelmonim Artoli. 2017. An Effective Deep Autoencoder Approach for Online Smartphone-Based Human Activity Recognition. *International Journal of Computer Science and Network Security* 17, 4 (2017), 160–165. http://paper.ijcsns.org/07_book/201704/20170423.pdf
- [6] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. 2018. Diagnosing Error in Temporal Action Detectors. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-030-01219-9_16
- [7] Behrooz Azadi, Michael Haslgrübler, Bernhard Anzengruber-Tanase, Georgios Sopidis, and Alois Ferscha. 2024. Robust Feature Representation Using Multi-Task Learning for Human Activity Recognition. *MDPI Sensors* 24, 2 (2024). <https://doi.org/10.3390/s24020681>

- [8] Lei Jimmy Ba and Rich Caruana. 2014. Do Deep Nets Really Need to Be Deep?. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/b0c355a9dedccb50e5537e8f2e3f0810-Paper.pdf
- [9] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. 2020. Boundary Content Graph Neural Network for Temporal Action Proposal Generation. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-030-58604-1_8
- [10] Ling Bao and Stephen S. Intille. 2004. Activity Recognition from User-Annotated Acceleration Data. In *International Conference on Pervasive Computing*. Springer. https://doi.org/10.1007/978-3-540-24646-6_1
- [11] Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent Is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166. <https://doi.org/10.1109/72.279181>
- [12] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R. Schmidtke, and Michael Beigl. 2010. ActiServ: Activity Recognition Service for Mobile Phones. In *International Symposium on Wearable Computers*. IEEE. <https://doi.org/10.1109/ISWC.2010.5665868>
- [13] Mehrab Bin Morshed, Harish Kashyap Haresamudram, Dheeraj Bandaru, Gregory D. Abowd, and Thomas Ploetz. 2022. A Personalized Approach for Developing a Snacking Detection System Using Earbuds in a Semi-Naturalistic Setting. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3544794.3558469>
- [14] Ulf Blanke and Bernt Schiele. 2009. Daily Routine Recognition through Activity Spotting. In *International Symposium on Location- and Context-Awareness*. Springer. https://doi.org/10.1007/978-3-642-01721-6_12
- [15] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. Tutorial on Deep Learning for Human Activity Recognition. *CoRR* abs/2110.06663 (2021). <https://doi.org/10.48550/arXiv.2110.06663>
- [16] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2022. Investigating (Re)Current State-of-the-art in Human Activity Recognition Datasets. *Frontiers in Computer Science* 4 (2022). <https://doi.org/10.3389/fcomp.2022.924954>

- [17] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. Improving Deep Learning for HAR with Shallow LSTMs. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3460421.3480419>
- [18] Marius Bock, Maximilian Hopp, Kristof Van Laerhoven, and Michael Moeller. 2025. Label Leakage in Federated Inertial-based Human Activity Recognition. *CoRR* abs/2505.20924 (2025). <https://doi.org/10.48550/arXiv.2505.20924>
- [19] Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. 2023. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. *CoRR* abs/2304.05088 (2023). <https://doi.org/10.48550/arXiv.2304.05088>
- [20] Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. 2024. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024). <https://doi.org/10.1145/3699776>
- [21] Marius Bock, Michael Moeller, and Kristof Van Laerhoven. 2025. DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activity Recognition. *CoRR* abs/2505.20894 (2025). <https://doi.org/10.48550/arXiv.2505.20894>
- [22] Marius Bock, Michael Moeller, and Kristof Van Laerhoven. 2024. Temporal Action Localization for Inertial-based Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024). <https://doi.org/10.1145/3699770>
- [23] Marius Bock, Kristof Van Laerhoven, and Michael Moeller. 2024. Weak-Annotation of HAR Datasets Using Vision Foundation Models. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3675095.3676613>
- [24] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering With Subgraph Embeddings. In *Conference on Empirical Methods in Natural Language Processing*. ACL. <https://doi.org/10.3115/v1/D14-1067>
- [25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. In *Advances in Neural Information Processing*

- Systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- [26] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *Comput. Surveys* 46, 3 (2014). <https://doi.org/10.1145/2499621>
 - [27] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr.2017.502>
 - [28] Shing Chan, Yuan Hang, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. 2024. CAPTURE-24: A Large Dataset of Wrist-Worn Activity Tracker Data Collected in the Wild for Human Activity Recognition. *Scientific Data* 11, 1 (2024). <https://doi.org/10.1038/s41597-024-03960-3>
 - [29] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. 2016. Fusion of Depth, Skeleton, and Inertial Data for Human Action Recognition. In *International Conference on Acoustics, Speech and Signal Processing*. IEEE. <https://doi.org/10.1109/ICASSP.2016.7472170>
 - [30] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. 2022. DCAN: Improving Temporal Action Detection via Dual Context Aggregation. In *AAAI Conference on Artificial Intelligence*. PKP. <https://doi.org/10.1609/aaai.v36i1.19900>
 - [31] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep Learning for Sensor-Based Human Activity Recognition: Overview, Challenges, and Opportunities. *Comput. Surveys* 54, 4 (2021). <https://doi.org/10.1145/3447744>
 - [32] Ling Chen, Rong Hu, Menghan Wu, and Xin Zhou. 2023. HMGAN: A Hierarchical Multi-Modal Generative Adversarial Network Model for Wearable Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023). <https://doi.org/10.1145/3610909>
 - [33] Ling Chen, Yi Zhang, Shenghuan Miao, Sirou Zhu, Rong Hu, Liangying Peng, and Mingqi Lv. 2023. SALIENCE: An Unsupervised User Adaptation Model for Multiple Wearable Sensors Based Human Activity Recognition. *IEEE Transactions on Mobile Computing* 22, 9 (2023), 5492–5503. <https://doi.org/10.1109/TMC.2022.3171312>

- [34] Ling Chen, Yi Zhang, and Liangying Peng. 2020. METIER: A Deep Multi-Task Learning Based Activity and User Recognition Model Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020). <https://doi.org/10.1145/3381012>
- [35] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. 2016. LSTM Networks for Mobile Human Activity Recognition. In *International Conference on Artificial Intelligence: Technologies and Applications*. Atlantic Press. <https://doi.org/10.2991/icaita-16.2016.13>
- [36] Feng Cheng and Gedas Bertasius. 2022. TallFormer: Temporal Action Localization With a Long-Memory Transformer. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-031-19830-4_29
- [37] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) From Scratch. *Journal of Machine Learning Research* 12, 76 (2011), 2493–2537. <http://jmlr.org/papers/v12/collobert11a.html>
- [38] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2022. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision* 130 (2022), 33–55. <https://doi.org/10.1007/s11263-021-01531-2>
- [39] de la Torre, Fernando, Jessica K. Hodgins, Javier Montano, and Sergio Valcarcel. 2009. Detailed Human Data Acquisition of Kitchen Activities: The CMU-Multimodal Activity Database (CMU-MMAC). In *Conference on Human Factors in Computing Systems*. ACM. http://www.cs.cmu.edu/~ftorre/web_page/humansensing.cs.cmu.edu/projects/CMU-MMAC.html
- [40] Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer. 2016. Monitoring Activities of Daily Living in Smart Homes: Understanding Human Behavior. *IEEE Signal Processing Magazine* 33, 2 (2016), 81–94. <https://doi.org/10.1109/MSP.2015.2503881>
- [41] Shohreh Deldari, Hao Xue, Aaqib Saeed, Daniel V. Smith, and Flora D. Salim. 2022. COCOA: Cross Modality Contrastive Learning for Sensor Data. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022). <https://doi.org/10.1145/3550316>

- [42] Joseph DelPreto, Chao Liu, Yiyue Luo, Michael Foshey, Yunzhu Li, Antonio Torralba, Wojciech Matusik, and Daniela Rus. 2022. ActionSense: A Multimodal Dataset and Recording Framework for Human Activities Using Wearable Sensors in a Kitchen Environment. In *International Conference on Neural Information Processing Systems*. Curran Associates Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/5985e81d65605827ac35401999aea22a-Paper-Datasets_and_Benchmarks.pdf
- [43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR.2009.5206848>
- [44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North*. ACL. <https://doi.org/10.18653/v1/N19-1423>
- [45] Alexander Diete and Heiner Stuckenschmidt. 2019. Fusing Object Information and Inertial Data for Activity Recognition. *MDPI Sensors* 19, 19 (2019), 4119. <https://doi.org/10.3390/s19194119>
- [46] Alexander Diete, Timo Sztyler, and Heiner Stuckenschmidt. 2019. Vision and Acceleration Modalities: Partners for Recognizing Complex Activities. In *International Conference on Pervasive Computing and Communications Workshops*. IEEE. <https://doi.org/10.1109/PERCOMW.2019.8730690>
- [47] Alexander Diete, Timo Sztyler, Lydia Weiland, and Heiner Stuckenschmidt. 2018. Improving Motion-Based Activity Recognition With Ego-Centric Vision. In *International Conference on Pervasive Computing and Communications Workshops*. IEEE. <https://doi.org/10.1109/PERCOMW.2018.8480334>
- [48] Muhammad Ehatisham-Ul-Haq, Ali Javed, Muhammad Awais Azam, Hafiz M. A. Malik, Aun Irtaza, Ik Hyun Lee, and Muhammad Tariq Mahmood. 2019. Robust Human Activity Recognition Using Multimodal Feature-Level Fusion. *IEEE Access: Practical innovations, Open Solutions* 7 (2019), 60736–60751. <https://doi.org/10.1109/ACCESS.2019.2913393>
- [49] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. 2023. Project Aria: A New Tool for Egocentric Multi-Modal AI Research. *CoRR* abs/2308.13561 (2023). <https://doi.org/10.48550/arXiv.2308.13561>

- [50] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2013. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1915–1929. <https://doi.org/10.1109/TPAMI.2012.231>
- [51] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/ICCV.2019.00630>
- [52] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M. P. Cardoso. 2010. Preprocessing Techniques for Context Recognition from Accelerometer Data. *Personal and Ubiquitous Computing* 14, 7 (2010). <https://doi.org/10.1007/s00779-010-0293-9>
- [53] Stefan Gerd Fritsch, Cennet Oguz, Vitor Fortes Rey, Lala Ray, Maximilian Kiefer-Emmanouilidis, and Paul Lukowicz. 2024. MuJo: Multimodal Joint Feature Space Learning for Human Activity Recognition. <https://doi.org/10.48550/arXiv.2406.03857>
- [54] Ziqi Gao, Yuntao Wang, Jianguo Chen, Junliang Xing, Shwetak Patel, Xin Liu, and Yuanchun Shi. 2023. MMTSA: Multi-Modal Temporal Segment Attention Network for Efficient Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023), 1–26. <https://doi.org/10.1145/3610872>
- [55] Marjan Ghazvininejad, Hamid R. Rabiee, Nima Pourdamghani, and Parisa Khanipour. 2011. HMM Based Semi-Supervised Learning for Activity Recognition. In *International Workshop on Situation Activity & Goal Awareness*. ACM. <https://doi.org/10.1145/2030045.2030065>
- [56] Rohit Girdhar and Kristen Grauman. 2021. Anticipative Video Transformer. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.01325>
- [57] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR. <http://proceedings.mlr.press/v9/glorot10a>
- [58] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. 2020. Scale Matters: Temporal Scale Aggregation Network for Precise Action Localization in Untrimmed Videos. In *International Conference on Multimedia and Expo*. IEEE. <https://doi.org/10.1109/ICME46284.2020.9102850>

- [59] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, and Xingyu Liu. 2022. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR52688.2022.01842>
- [60] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, et al. 2024. Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives. In *Conference on Computer Vision and Pattern Recognition*. IEEE. https://openaccess.thecvf.com/content/CVPR2024/html/Grauman_Ego-Exo4D_Understanding_Skilled_Human_Activity_from_First-_and_Third-Person_Perspectives_CVPR_2024_paper.html
- [61] Yu Guan and Thomas Plötz. 2017. Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017). <https://doi.org/10.1145/3090076>
- [62] Sojeong Ha and Seungjin Choi. 2016. Convolutional Neural Networks for Human Activity Recognition Using Multiple Accelerometer and Gyroscope Sensors. In *International Joint Conference on Neural Networks*. IEEE. <https://doi.org/10.1109/IJCNN.2016.7727224>
- [63] Sojeong Ha, Jeong-Min Yun, and Seungjin Choi. 2015. Multi-Modal Convolutional Neural Networks for Activity Recognition. In *International Conference on Systems, Man, and Cybernetics*. IEEE. <https://doi.org/10.1109/SMC.2015.525>
- [64] Nils Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Ploetz. 2015. PD Disease State Assessment in Naturalistic Environments Using Deep Learning. In *AAAI Conference on Artificial Intelligence*. PKP. <https://ojs.aaai.org/index.php/AAAI/article/view/9484>
- [65] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In *International Joint Conference on Artificial Intelligence*. <https://doi.org/10.5555/3060832.3060835>
- [66] Nils Y. Hammerla and Thomas Plötz. 2015. Let’s (Not) Stick Together: Pairwise Similarity Biases Cross-Validation in Activity Recognition. In *International Joint Confer-*

- ence on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/2750858.2807551>
- [67] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L. Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. 2020. Masked Reconstruction Based Self-Supervision for Human Activity Recognition. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3410531.3414306>
 - [68] Harish Haresamudram, Chi Ian Tang, Sungho Suh, Paul Lukowicz, and Thomas Ploetz. 2023. Solving the Sensor-based Activity Recognition Problem (SOAR): Self-supervised, Multi-modal Recognition of Activities from Wearable Sensors. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/3594739.3605102>
 - [69] Harish Haresamudram, Chi Ian Tang, Sungho Suh, Paul Lukowicz, and Thomas Plötz. 2024. Solving the Sensor-Based Activity Recognition Problem (SOAR): Self-Supervised, Multi-Modal Recognition of Activities from Wearable Sensors. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/3675094.3677562>
 - [70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr.2016.90>
 - [71] Fabian Caba Heilbron, Juan Carlos Niebles, Victor Escorcia, and Bernard Ghanem. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr.2015.7298698>
 - [72] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97. <https://doi.org/10.1109/MSP.2012.2205597>
 - [73] Shruthi Kashinath Hiremath and Thomas Ploetz. 2021. On the Role of Context Length for Feature Extraction and Sequence Modeling in Human Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/3460421.3478825>

- [74] Tom Hoddes, Alex Bijamov, Saket Joshi, Daniel Roggen, Ali Etemad, Robert Harle, and David Racz. 2025. Scaling Laws in Wearable Human Activity Recognition. *CoRR* abs/2502.03364 (2025). <https://doi.org/10.48550/arXiv.2502.03364>
- [75] Alexander Hoelzemann, Marius Bock, Ericka Andrea Valladares Bastías, Salma El Ouazzani Touhami, Kenza Nassiri, and Kristof Van Laerhoven. 2023. A Data-Driven Study on the Hawthorne Effect in Sensor-Based Human Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/3594739.3610743>
- [76] Alexander Hoelzemann, Marius Bock, and Kristof Van Laerhoven. 2024. Evaluation of Video-Assisted Annotation of Human IMU Data Across Expertise, Datasets, and Tools. In *International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events*. IEEE. <https://doi.org/10.1109/percomworkshops59983.2024.10503292>
- [77] Alexander Hoelzemann, Julia Lee Romero, Marius Bock, Kristof Van Laerhoven, and Qin Lv. 2023. Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors. *Sensors* 23, 13 (2023). <https://doi.org/10.3390/s23135879>
- [78] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The Dataset Nutrition Label: A Framework to Drive Higher Data Quality Standards. *CoRR* abs/1805.03677 (2018). <https://arxiv.org/abs/1805.03677>
- [79] Menghao Hu, Mingxuan Luo, Menghua Huang, Wenhua Meng, Baochen Xiong, Xiaoshan Yang, and Jitao Sang. 2023. Towards a Multimodal Human Activity Dataset for Healthcare. *Multimedia Systems* 29, 1 (2023), 1–13. <https://doi.org/10.1007/s00530-021-00875-6>
- [80] Momal Ijaz, Renato Diaz, and Chen Chen. 2022. Multimodal Transformer for Nursing Activity Recognition. In *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. <https://doi.org/10.1109/cvprw56347.2022.00224>
- [81] Javed Imran and Balasubramanian Raman. 2020. Evaluating Fusion of RGB-D and Inertial Sensors for Multimodal Human Action Recognition. *Journal of Ambient Intelligence and Humanized Computing* 11, 1 (2020), 189–208. <https://doi.org/10.1007/s12652-019-01239-9>

- [82] Javed Imran and Balasubramanian Raman. 2020. Multimodal Egocentric Activity Recognition Using Multi-Stream CNN. In *Indian Conference on Computer Vision, Graphics and Image Processing*. ACM. <https://doi.org/10.1145/3293353.3293363>
- [83] Sozo Inoue, Paula Lago, Tahera Hossain, Tittaya Mairittha, and Nattaya Mairittha. 2019. Integrating Activity Recognition and Nursing Care Records: The System, Deployment, and a Verification Study. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019). <https://doi.org/10.1145/3351244>
- [84] Md Mofijul Islam and Tariq Iqbal. 2021. Multi-GAT: A Graphical Attention-Based Hierarchical Multimodal Representation Learning Approach for Human Activity Recognition. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1729–1736. <https://doi.org/10.1109/LRA.2021.3059624>
- [85] Md Mofijul Islam and Tariq Iqbal. 2022. MuMu: Cooperative Multitask Learning-Based Guided Multimodal Fusion. In *AAAI Conference on Artificial Intelligence*. PKP. <https://doi.org/10.1609/aaai.v36i1.19988>
- [86] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. ACL. <https://doi.org/10.3115/v1/P15-1001>
- [87] Wenchao Jiang and Zhaozheng Yin. 2015. Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In *International Conference on Multimedia*. ACM. <https://doi.org/10.1145/2733373.2806333>
- [88] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Suktankar. 2014. THUMOS Challenge: Action Recognition With a Large Number of Classes. <http://crcv.ucf.edu/THUMOS14/>
- [89] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and Understanding Recurrent Networks. *CoRR* abs/1506.02078 (2015). <https://doi.org/10.48550/arXiv.1506.02078>
- [90] 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. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017). <https://doi.org/10.48550/arXiv.1705.06950>

- [91] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. 2021. Contrastive Self-supervised Learning for Sensor-based Human Activity Recognition. In *International Joint Conference on Biometrics*. IEEE. <https://doi.org/10.1109/IJCB52358.2021.9484410>
- [92] Tae-Young Kim and Sung-Bae Cho. 2019. Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks. *Energy* 182 (2019), 72–81. <https://doi.org/10.1016/j.energy.2019.05.230>
- [93] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*. <https://doi.org/10.48550/arXiv.2010.11929>
- [94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *International Conference on Neural Information Processing Systems*. <https://doi.org/10.5555/2999134.2999257>
- [95] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. 2019. Unsupervised Learning of Action Classes With Continuous Temporal Embedding. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR.2019.01234>
- [96] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. 2020. IMUTube: Automatic Extraction of Virtual On-Body Accelerometry From Video for Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–29. <https://doi.org/10.1145/3411841>
- [97] Hyeokhyen Kwon, Bingyao Wang, Gregory D. Abowd, and Thomas Plötz. 2021. Approaching the Real-World: Supporting Activity Recognition Training with Virtual IMU Data. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021). <https://doi.org/10.1145/3478096>
- [98] Nicholas D. Lane and Petko Georgiev. 2015. Can Deep Learning Revolutionize Mobile Sensing?. In *16th International Workshop on Mobile Computing Systems and Applications*. ACM. <https://doi.org/10.1145/2699343.2699349>

- [99] Song-Mi Lee, Sang M. Yoon, and Heeryon Cho. 2017. Human Activity Recognition From Accelerometer Data Using Convolutional Neural Network. In *International Conference on Big Data and Smart Computing*. IEEE, 131–134. <https://doi.org/10.1109/BIGCOMP.2017.7881728>
- [100] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 2020. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/f0bda020d2470f2e74990a07a607ebd9-Paper.pdf
- [101] Yongmou Li, Dianxi Shi, Bo Ding, and Dongbo Liu. 2014. Unsupervised Feature Learning for Human Activity Recognition Using Smartphone Sensors. In *Mining Intelligence and Knowledge Exploration*. Springer. https://doi.org/10.1007/978-3-319-13817-6_11
- [102] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection. In *Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr52688.2022.00476>
- [103] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. 2020. Fast Learning of Temporal Action Proposal via Dense Boundary Generator. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1609/aaai.v34i07.6815>
- [104] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. 2021. Learning Salient Boundary Feature for Anchor-Free Temporal Action Localization. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr46437.2021.00333>
- [105] Chi-Heng Lin, Chiraag Kaushik, Eva L. Dyer, and Vidya Muthukumar. 2024. The Good, the Bad and the Ugly Sides of Data Augmentation: An Implicit Spectral Regularization Perspective. *Journal of Machine Learning Research* 25, 91 (2024), 1–85. <http://jmlr.org/papers/v25/22-1312.html>
- [106] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv.2019.00399>

- [107] Qinying Liu and Zilei Wang. 2020. Progressive Boundary Refinement Network for Temporal Action Detection. In *AAAI Conference on Artificial Intelligence*. PKP. <https://doi.org/10.1609/aaai.v34i07.6829>
- [108] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. GlobalFusion: A Global Attentional Deep Learning Framework for Multisensor Information Fusion. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27. <https://doi.org/10.1145/3380999>
- [109] Xiaolong Liu, Song Bai, and Xiang Bai. 2022. An Empirical Study of End-To-End Temporal Action Detection. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr52688.2022.01938>
- [110] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip H. S. Torr. 2021. Multi-Shot Temporal Event Localization: A Benchmark. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr46437.2021.01241>
- [111] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. 2022. End-To-End Temporal Action Detection With Transformer. *IEEE Transactions on Image Processing* 31 (2022), 5427–5441. <https://doi.org/10.1109/TIP.2022.3195321>
- [112] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.00986>
- [113] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video Swin Transformer. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr52688.2022.00320>
- [114] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. 2019. Gaussian Temporal Awareness Networks for Action Localization. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr.2019.00043>
- [115] Yantao Lu and Senem Velipasalar. 2018. Human Activity Classification Incorporating Egocentric Video and Inertial Measurement Unit Data. In *Global Conference on Signal and Information Processing*. IEEE. <https://doi.org/10.1109/GlobalSIP.2018.8646367>

- [116] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval and Captioning. *Neurocomputing* 508 (2022). <https://doi.org/10.1016/j.neucom.2022.07.028>
- [117] Haojie Ma, Zhijie Zhang, Wenzhong Li, and Sanglu Lu. 2021. Unsupervised Human Activity Representation Learning with Multi-task Deep Clustering. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021). <https://doi.org/10.1145/3448074>
- [118] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Can Gradient Clipping Mitigate Label Noise?. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=rklB76EKPr>
- [119] Shenghuan Miao, Ling Chen, Rong Hu, and Yingsong Luo. 2022. Towards a Dynamic Inter-Sensor Correlations Learning Framework for Multi-Sensor-Based Wearable Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–25. <https://doi.org/10.1145/3550331>
- [120] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. 2011. Strategies for Training Large Scale Neural Network Language Models. In *Workshop on Automatic Speech Recognition & Understanding*. IEEE. <https://doi.org/10.1109/ASRU.2011.6163930>
- [121] Vishvak S. Murahari and Thomas Plötz. 2018. On Attention Models for Human Activity Recognition. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3267242.3267287>
- [122] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. 2022. Proposal-Free Temporal Action Detection via Global Segmentation Mask Learning. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-031-20062-5_37
- [123] Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep Learning Applications and Challenges in Big Data Analytics. *Journal of Big Data* 2, 1 (2015). <https://doi.org/10.1186/s40537-014-0007-7>
- [124] Katsuyuki Nakamura, Serena Yeung, Alexandre Alahi, and Li Fei-Fei. 2017. Jointly Learning Energy Expenditures and Activities Using Egocentric Multimodal Signals. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR.2017.721>

- [125] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *CoRR* abs/2304.07193 (2024). <https://doi.org/10.48550/arXiv.2304.07193>
- [126] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *MDPI Sensors* 16, 1 (2016). <https://doi.org/10.3390/s16010115>
- [127] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *International Conference on Machine Learning (Proceedings of Machine Learning Research)*. PMLR. <https://proceedings.mlr.press/v28/pascanu13.html>
- [128] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. 2005. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In *International Symposium on Wearable Computers*. IEEE. <https://doi.org/10.1109/ISWC.2005.22>
- [129] Lloyd Pellatt, Marius Bock, Daniel Roggen, and Kristof Van Laerhoven. 2022. A Public Repository to Improve Replicability and Collaboration in Deep Learning for HAR. In *International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events*. IEEE. <https://doi.org/10.1109/PerComWorkshops53856.2022.9767436>
- [130] Lloyd Pellatt and Daniel Roggen. 2020. CausalBatch: Solving Complexity/Performance Tradeoffs for Deep Convolutional and Lstm Networks for Wearable Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing and International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3410530.3414365>
- [131] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–16. <https://doi.org/10.1145/3214277>

- [132] Rafael Possas, Sheila Pinto Caceres, and Fabio Ramos. 2018. Egocentric Activity Recognition on a Budget. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR.2018.00625>
- [133] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria P. Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S. Iyengar. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *Comput. Surveys* 51, 5 (2018). <https://doi.org/10.1145/3234150>
- [134] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. 2021. Temporal Context Aggregation Network for Temporal Action Proposal Refinement. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr46437.2021.00055>
- [135] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). <https://doi.org/10.48550/arXiv.2103.00020>
- [136] Valentin Radu and Maximilian Henne. 2019. Vision2Sensor: Knowledge Transfer Across Sensing Modalities for Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019). <https://doi.org/10.1145/3351242>
- [137] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. 2005. Activity Recognition From Accelerometer Data. In *Conference on Innovative Applications of Artificial Intelligence*. ACM. <https://dl.acm.org/doi/10.5555/1620092.1620107>
- [138] Vitor Fortes Rey, Peter Hevesi, Onorina Kovalenko, and Paul Lukowicz. 2019. Let There Be IMU Data: Generating Training Data for Wearable, Motion Sensor Based Activity Recognition from Monocular RGB Videos. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/3341162.3345590>
- [139] Vitor Fortes Rey, Lala Shakti Swarup Ray, Xia Qingxin, Kaishun Wu, and Paul Lukowicz. 2024. Enhancing Inertial Hand Based HAR through Joint Representation of Language, Pose and Synthetic IMUs. <https://doi.org/10.48550/ARXIV.2406.01316>

- [140] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171 (2016). <https://doi.org/10.1016/j.neucom.2015.07.085>
- [141] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr.2019.00075>
- [142] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkel, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting Complex Activity Datasets in Highly Rich Networked Sensor Environments. In *International Conference on Networked Sensing Systems*. IEEE. <https://doi.org/10.1109/INSS.2010.5573462>
- [143] Charissa A. Ronao and Sung-Bae Cho. 2016. Human Activity Recognition With Smartphone Sensors Using Deep Learning Neural Networks. *Expert Systems with Applications* 59 (2016), 235–244. <https://doi.org/10.1016/j.eswa.2016.04.032>
- [144] Debaditya Roy and Basura Fernando. 2022. Predicting the Next Action by Modeling the Abstract Goal. *CoRR* abs/2209.05044 (2022). <https://doi.org/10.48550/arxiv.2209.05044>
- [145] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-Task Self-Supervised Learning for Human Activity Detection. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019). <https://doi.org/10.1145/3328932>
- [146] Tara N. Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. 2013. Deep Convolutional Neural Networks for LVCSR. In *International Conference on Acoustics, Speech and Signal Processing*. IEEE. <https://doi.org/10.1109/ICASSP.2013.6639347>
- [147] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. 2013. TV-L1 Optical Flow Estimation. *Image Processing On Line* 3 (2013), 137–150. <https://doi.org/10.5201/ipol.2013.26>
- [148] Philipp M. Scholl, Benjamin Völker, Bernd Becker, and Kristof Van Laerhoven. 2019. A Multi-Media Exchange Format for Time-Series Dataset Curation. In *Human Activity Sensing*. Springer, 111–119. https://doi.org/10.1007/978-3-030-13001-5_8

- [149] Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. 2015. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. <https://doi.org/10.1145/2750858.2807547>
- [150] Shuai Shao, Yu Guan, and Victor Sanchez. 2024. Beyond Isolated Frames: Enhancing Sensor-Based Human Activity Recognition through Intra- and Inter-Frame Attention. *CoRR* abs/2405.19349 (2024). <https://doi.org/10.48550/arXiv.2405.19349> arXiv:2405.19349 [eess]
- [151] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. TriDet: Temporal Action Detection With Relative Boundary Modeling. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr52729.2023.01808>
- [152] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. 2022. ReAct: Temporal Action Detection With Relational Queries. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-031-20080-9_7
- [153] Sibong Song, Vijay Chandrasekhar, Bappaditya Mandal, Liyuan Li, Joo-Hwee Lim, Giduthuri Sateesh Babu, Phyo Phyo San, and Ngai-Man Cheung. 2016. Multimodal Multi-Stream Deep Learning for Egocentric Activity Recognition. In *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. <https://doi.org/10.1109/CVPRW.2016.54>
- [154] Sibong Song, Ngai-Man Cheung, Vijay Chandrasekhar, Bappaditya Mandal, and Jie Lin. 2016. Egocentric Activity Recognition With Multimodal Fisher Vector. In *International Conference on Acoustics, Speech and Signal Processing*. IEEE. <https://doi.org/10.1109/icassp.2016.7472171>
- [155] Ekaterina H. Spriggs, Fernando De La Torre, and Martial Hebert. 2009. Temporal Segmentation and Activity Classification From First-Person Sensing. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. <https://doi.org/10.1109/CVPRW.2009.5204354>
- [156] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaixin Li, Peng Dai, and Juwei Lu. 2021. Class Semantics-Based Attention for Action Detection. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.01348>

- [157] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. 2011. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2011), 2521–2537. <https://doi.org/10.1109/tpami.2011.36>
- [158] Maja Stikic, Diane Larlus, and Bernt Schiele. 2009. Multi-Graph Based Semi-supervised Learning for Activity Recognition. In *International Symposium on Wearable Computers*. IEEE. <https://doi.org/10.1109/ISWC.2009.24>
- [159] Maja Stikic and Bernt Schiele. 2009. Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning. In *International Symposium on Location- and Context-Awareness*. Springer. https://doi.org/10.1007/978-3-642-01721-6_10
- [160] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor S. Prentow, Mikkel B. Kjærgaard, Anind Dey, Tobias Sonne, and Mads M. Jensen. 2015. Smart Devices Are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In *Conference on Embedded Networked Sensor Systems*. ACM. <https://doi.org/10.1145/2809695.2809718>
- [161] David Strömbäck, Sangxia Huang, and Valentin Radu. 2020. MM-Fit: Multimodal Deep Learning for Automatic Exercise Logging across Sensing Devices. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020). <https://doi.org/10.1145/3432701>
- [162] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *International Conference on Neural Information Processing Systems*. ACM. <https://dl.acm.org/doi/10.5555/2969033.2969173>
- [163] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/CVPR.2015.7298594>
- [164] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-Body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *International Conference on Pervasive Computing and Communications*. IEEE. <https://doi.org/10.1109/PERCOM.2016.7456521>
- [165] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. 2021. Relaxed Transformer Decoders for Direct Action Proposal Generation. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.01327>

- [166] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. 2021. Exploring Contrastive Learning in Human Activity Recognition for Healthcare. *CoRR* abs/2011.11542 (2021). <https://doi.org/10.48550/arXiv.2011.11542> arXiv:2011.11542 [cs, eess]
- [167] Tuan N. Tang, Kwonyoung Kim, and Kwanghoon Sohn. 2023. TemporalMaxer: Maximize Temporal Context With Only Max Pooling for Temporal Action Localization. *CoRR* abs/2303.09055 (2023). <https://doi.org/10.48550/arXiv.2303.09055>
- [168] Jafar Tanha, Maarten Van Someren, and Hamideh Afsarmanesh. 2017. Semi-Supervised Self-Training for Decision Tree Classifiers. *International Journal of Machine Learning and Cybernetics* 8, 1 (2017). <https://doi.org/10.1007/s13042-015-0328-7>
- [169] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-030-58536-5_24
- [170] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. 2014. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *International Conference on Neural Information Processing Systems*. ACM. <https://dl.acm.org/doi/10.5555/2968826.2969027>
- [171] Catherine Tong, Jinchen Ge, and Nicholas D. Lane. 2021. Zero-Shot Learning for IMU-Based Activity Recognition Using Video Embeddings. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–23. <https://doi.org/10.1145/3494995>
- [172] M Trumble, A Gilbert, C Malleson, A Hilton, and J Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *British Machine Vision Conference*. BMVA Press. <https://doi.org/10.5244/c.31.14>
- [173] Warren W. Tryon. 2006. Activity Measurement. In *Clinician’s Handbook of Adult Behavioral Assessment*. Academic Press, 85–120. <https://doi.org/10.1016/B978-012343013-7/50006-9>
- [174] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. 2018. Context Recognition In-The-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018). <https://doi.org/10.1145/3161192>

- [175] Kristof Van Laerhoven, Alexander Hoelzemann, Iris Pahmeier, Andrea Teti, and Lars Gabrys. 2022. Validation of an Open-Source Ambulatory Assessment System in Support of Replicable Activity Studies. *German Journal of Exercise and Sport Research* 52, 2 (2022), 262–272. <https://doi.org/10.1007/s12662-022-00813-2>
- [176] Kristof Van Laerhoven, David Kilian, and Bernt Schiele. 2008. Using Rhythm Awareness in Long-Term Activity Recognition. In *International Symposium on Wearable Computers*. IEEE. <https://doi.org/10.1109/ISWC.2008.4911586>
- [177] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. <https://doi.org/10.48550/arXiv.1706.03762>
- [178] Rosaura G. VidalMata, Walter J. Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. 2021. Joint Visual-Temporal Embedding for Unsupervised Learning of Actions in Untrimmed Sequences. In *Winter Conference on Applications of Computer Vision*. IEEE. <https://doi.org/10.1109/wacv48630.2021.00128>
- [179] Aiguo Wang, Guilin Chen, Cuijuan Shang, Miaofei Zhang, and Li Liu. 2016. Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders. In *International Conference on Web-Age Information Management*. Springer. https://doi.org/10.1007/978-3-319-47121-1_3
- [180] Mengmeng Wang, Jiazheng Xing, and Yong Liu. 2021. ActionCLIP: A New Paradigm for Video Action Recognition. *CoRR* abs/2109.08472 (2021). <https://doi.org/10.48550/arXiv.2109.08472>
- [181] Jamie A. Ward, Paul Lukowicz, and Hans W. Gellersen. 2011. Performance Metrics for Activity Recognition. *ACM Transactions on Intelligent Systems and Technology* 2, 1 (2011). <https://doi.org/10.1145/1889681.1889687>
- [182] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. 2006. Evaluating Performance in Continuous Context Recognition Using Event-Driven Error Characterisation. In *International Symposium on Location- and Context-Awareness*. Springer. https://doi.org/10.1007/11752967_16
- [183] Jamie A. Ward, Paul Lukowicz, Gerhard Tröster, and Thad E. Starner. 2006. Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2006), 1553–1567. <https://doi.org/10.1109/TPAMI.2006.197>

- [184] Haoran Wei and Nasser Kehtarnavaz. 2020. Simultaneous Utilization of Inertial and Video Sensing for Action Detection and Recognition in Continuous Action Streams. *IEEE Sensors Journal* 20, 11 (2020), 6055–6063. <https://doi.org/10.1109/JSEN.2020.2973361>
- [185] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A Discriminative Feature Learning Approach for Deep Face Recognition. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-319-46478-7_31
- [186] Christoph Wieland and Victor Pankratius. 2023. TinyGraphHAR: Enhancing Human Activity Recognition With Graph Neural Networks. In *World AI IoT Congress*. IEEE. <https://doi.org/10.1109/AIIoT58121.2023.10174597>
- [187] Tz-Ying Wu, Ting-An Chien, Cheng-Sheng Chan, Chan-Wei Hu, and Min Sun. 2017. Anticipating Daily Intention Using On-Wrist Motion Triggered Sensing. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/ICCV.2017.15>
- [188] Rui Xi, Mengshu Hou, Mingsheng Fu, Hong Qu, and Daibo Liu. 2018. Deep Dilated Convolution on Multimodality Time Series for Human Activity Recognition. In *International Joint Conference on Neural Networks*. IEEE. <https://doi.org/10.1109/IJCNN.2018.8489540>
- [189] Kun Xia, Jianguang Huang, and Hanyu Wang. 2020. LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access* 8 (2020). <https://doi.org/10.1109/ACCESS.2020.2982225>
- [190] Kang Xia, Wenzhong Li, Shiwei Gan, and Sanglu Lu. 2023. TS2ACT: Few-Shot Human Activity Sensing with Cross-Modal Co-Learning. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 4 (2023). <https://doi.org/10.1145/3631445>
- [191] Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. 2019. InnoHAR: A Deep Neural Network for Complex Human Activity Recognition. *IEEE Access* 7 (2019). <https://doi.org/10.1109/ACCESS.2018.2890675>
- [192] Linfeng Xu, Qingbo Wu, Lili Pan, Fanman Meng, Hongliang Li, Chiyuan He, Hanxin Wang, Shaoxu Cheng, and Yu Dai. 2024. Towards Continual Egocentric Activity Recognition: A Multi-Modal Egocentric Activity Dataset for Continual Learning. *IEEE Transactions on Multimedia* 26 (2024), 2430–2443. <https://doi.org/10.1109/TMM.2023.3295899>

- [193] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr42600.2020.01017>
- [194] Jian B. Yang, Minh N. Nguyen, Phyo P. San, Xiao L. Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *International Joint Conference on Artificial Intelligence*. ACM. <https://doi.org/10.5555/2832747.2832806>
- [195] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. 2023. BasicTAD: An Astounding RGB-Only Baseline for Temporal Action Detection. *Computer Vision and Image Understanding* 232 (2023). <https://doi.org/10.1016/j.cviu.2023.103692>
- [196] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing. In *International Conference on World Wide Web*. ACM. <https://doi.org/10.1145/3038912.3052577>
- [197] Haibin Yu, Guoxiong Pan, Mian Pan, Chong Li, Wenyan Jia, Li Zhang, and Mingui Sun. 2019. A Hierarchical Deep Fusion Framework for Egocentric Activity Recognition Using a Wearable Hybrid Sensor System. *MDPI Sensors* 19, 3 (2019). <https://doi.org/10.3390/s19030546>
- [198] Hang Yuan, Shing Chan, Andrew P. Creagh, Catherine Tong, Aidan Acquah, David A. Clifton, and Aiden Doherty. 2024. Self-Supervised Learning for Human Activity Recognition Using 700,000 Person-Days of Wearable Data. *npj Digital Medicine* 7, 1 (2024), 91. <https://doi.org/10.1038/s41746-024-01062-3>
- [199] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-To-Token ViT: Training Vision Transformers From Scratch on Imagenet. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.00060>
- [200] Yuta Yuki, Junto Nozaki, Kei Hiroi, Katsuhiko Kaji, and Nobuo Kawaguchi. 2018. Activity Recognition Using Dual-ConvLstm Extracting Local and Global Features for Shl Recognition Challenge. In *International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM. <https://doi.org/10.1145/3267305.3267533>

- [201] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. 2008. Activity Recognition From On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection. In *European Conference on Wireless Sensor Networks*. Springer. https://doi.org/10.1007/978-3-540-77690-1_2
- [202] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. 2019. Graph Convolutional Networks for Temporal Action Localization. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv.2019.00719>
- [203] Bing Zhai, Yu Guan, Michael Catt, and Thomas Plötz. 2021. Ubi-SleepNet: Advanced Multimodal Fusion Techniques for Three-stage Sleep Classification Using Ubiquitous Sensing. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021). <https://doi.org/10.1145/3494961>
- [204] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing Moments of Actions With Transformers. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-031-19772-7_29
- [205] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A Framework for Human Activity Recognition Using Imu Fusion and Convtransformer. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022). <https://doi.org/10.1145/3534584>
- [206] Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/f2925f97bc13ad2852a7a551802f000-Paper.pdf
- [207] Chen Zhao, Ali K. Thabet, and Bernard Ghanem. 2021. Video Self-Stitching Graph Network for Temporal Action Localization. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.01340>
- [208] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Bottom-up Temporal Action Localization With Mutual Regularization. In *European Conference on Computer Vision*. Springer. https://doi.org/10.1007/978-3-030-58598-3_32

- [209] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM Neural Network for Text Classification. *CoRR* abs/1511.08630 (2015). <https://doi.org/10.48550/arXiv.1511.08630>
- [210] Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. 2022. TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In *International Symposium on Wearable Computers*. ACM. <https://doi.org/10.1145/3544794.3558467>
- [211] Yexu Zhou, Haibin Zhao, Yiran Huang, Tobias Röddiger, Murat Kurnaz, Till Riedel, and Michael Beigl. 2024. AutoAugHAR: Automated Data Augmentation for Sensor-based Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024). <https://doi.org/10.1145/3659589>
- [212] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. 2021. Enriching Local and Global Contexts for Temporal Action Localization. In *International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv48922.2021.01326>

A. Supplementary Material: WEAR An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition

A.1. Methodological Transparency Appendix & License

The source code that was used to conduct all experiments, postprocessing and evaluation steps mentioned in this paper is available via Github (mariusbock.github.io/wear/). The repository is written in such a way that other architectures (both inertial- and vision-based) can be added in the future. The repository provides Readme files which give details on the overall structure of the repository, how collect additional data and how to set up an Anaconda environment with the needed packages to run experiments. Experiments are defined via '.json'-format configuration files which allow for easy sharing of used hyperparameter settings. Experiments were conducted on a cluster-based system, with each run being assigned one Tesla V100 GPU, 10 GB of RAM and two AMD EPYC 7452 CPUs.

WEAR and all associated files are offered under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. The complete dataset can be downloaded either via sciebo (https://bit.ly/wear_dataset) or a server hosted by the Ubiquitous Computing group at the University of Siegen (https://bit.ly/wear_data). The dataset download is structured into the (1) '.json'-formatted annotations, (2) raw, synchronized inertial and vision data and (3) precomputed feature embeddings as mentioned in the main paper. This file is current as of submission of this dissertation. Please refer to the GitHub repository (mariusbock.github.io/wear/) to access the latest version the supplementary material as well as the WEAR dataset.

A.2. Dataset Overview and Contents

The outdoor sports dataset WEAR features data of 22 participants performing each a total of 18 different workout activities with untrimmed inertial (acceleration) and camera (egocentric video) data recorded at 11 different outside locations. It provides a challenging prediction scenario marked by purposely introduced activity variations and an overall small information overlap across modalities. Figure A.1 provides a dataset nutrition label inspired by Holland et al. [78] in a table-like manner.

WEAR Dataset Key Facts	
Motivation	An outdoor sports dataset (egocentric-video & inertial data) with small information overlap across modalities
Example Use Cases	Inertial-based, vision-based & multimodal Human Activity Recognition
Authors	Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, Michael Moeller
Meta information	
Dataset	
Locations	11
Activities	18 workout activities + NULL-class
Action segments	751
Total duration	1137 min
Per-participant	47 ± 11 min
Subjects	
Count	22 (13 male, 9 female)
Age	27.59 ± 4.69
Height	175.2 ± 9.83 cm
Weight	69.74 ± 11.23 kg
Modalities	
3D-Acceleration	
Sensor	Bangle.js Version 1
Settings	50 Hz (± 8g)
Format	.csv
Placement	Ankles & Wrists
Size	811.50 MB
Egocentric Vision	
Sensor	GoPro Hero 8 & 11
Settings	1080p 60 FPS; SuperView FOV
Format	.mp4
Placement	Head (tilted 25° downwards)
Size	174,74 GB

Figure A.1.: Dataset nutrition label of the WEAR dataset. The dataset nutrition label was originally proposed by Holland et al. [78]. Our adaptation is inspired by DelPreto et al. [42].

A.2.1. Intended Uses and Ethical Considerations

Before participating in the study, participants were notified that by nature the data they provide can only be pseudonymised. This means that, though requiring a substantial amount of effort, the identity of a person can be reconstructed. Although participants agreed to include their egocentric videos in a public dataset, it is essential to refrain from actively identifying the individuals featured in the WEAR dataset. If other researchers decide to contribute to the WEAR dataset by recording additional participants, societal and ethical implications should be considered. As with the participants part of the original release of the WEAR dataset, all participants must be briefed before their first recording, making them aware of all necessary information and implications that come with providing to the WEAR dataset. Recording locations should only be chosen if video recordings are allowed at said location and participants are given enough space to perform each activity safely. If the recording location involves pedestrians walking within close proximity, pedestrians should be notified that they are being recorded and, if applicable, captured faces should be blurred during postprocessing.

The WEAR dataset and associated code are made public for research purposes. With the accurate detection of physical activities that we perform in our daily lives having been identified as valuable information, the WEAR dataset focuses on one of the most popular application scenarios of wearable smartwatches and action cameras, i.e. self-tracking of workout activities. With the ease of reproducibility we hope to make WEAR a collaborative, expanding dataset which researchers from different locations and backgrounds can contribute to. For example, as the current selection of participants is biased towards healthy, young people, we hope to overcome said limitation by including people from more diverse backgrounds and age groups in future iterations of the dataset.

Lastly, the authors took great care of avoiding any infringement of rights during the data collection process. Yet, in case of conflicts, they are of course committed to taking appropriate actions, such as promptly removing data associated with such concerns.

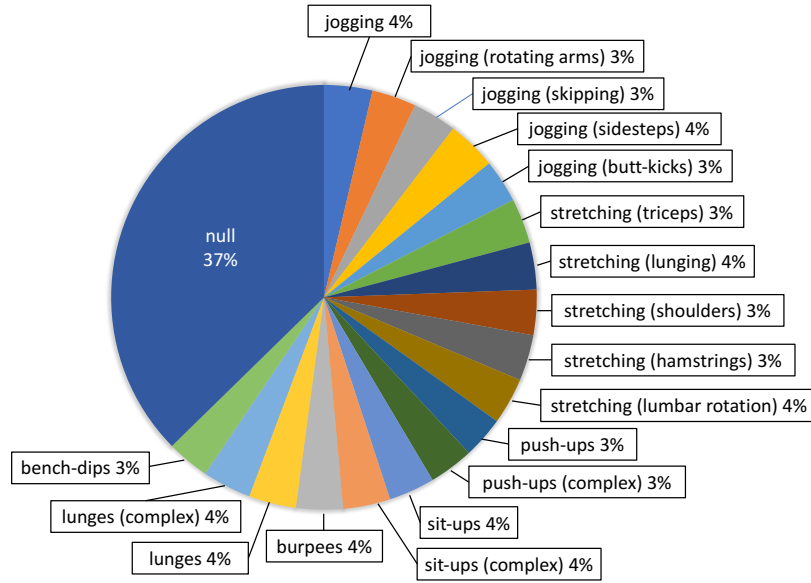


Figure A.2.: Overview of the activity classes featured in the WEAR dataset. Percentages are measured relative to the total size of the dataset (around 1,200 minutes). The null class refers to samples not belonging to any of the classes of interest. A detailed description of each activity can be found in the recording plan attached at the end of the supplementary material.

A.2.2. Class Distribution

Figure A.2 gives an overview of the 18 activity classes featured in the WEAR dataset and provides number of coherent sequences as well as total duration per workout activity class.

A.2.3. Participant and Session Information

The location and the time of day at which the sessions were performed were not fixed and thus vary across subjects. As participants were allowed to split activities across more (or less) than two sessions, session counts vary across subjects. Table A.1 provides information on all 11 recording locations that are part of the WEAR dataset. The table details general information such as surface conditions of the location as well as which direction the static camera seen in videos is facing. Table A.2 provides supplementary information on all separate sessions contained in the dataset. For each session, we detail its overall length in minutes, the number of distinct activities performed by the participant, the location it was recorded at, the month and time of day it was recorded, as well as the overall weather conditions during the duration of the session.

After having completed all sessions, participants were asked to take part in a questionnaire which was used to gather vital information (gender, age, height and weight) as well as workout-specific questions, aiming towards assessing the overall fitness level and experience

Table A.1.: Description of the 11 locations featured in the WEAR dataset. For each location we provide information on surface conditions, overall surroundings and direction the static camera is facing.

Location ID	Description
1	Meadow in proximity to a larger building. Area is surrounded by trees with from November on-wards, fallen leaves laying on the ground. Static camera faces North-West.
2	Parking lot in proximity to building. Concrete surface. Static camera faces West.
3	Small square with concrete surface. Surrounded by bushes and buildings. Static camera faces West.
4	Meadow enclosed by bungalow-style living quarters. Static camera faces North-East
5	Covered walkway next to a building. Concrete surface. Walkway enclosed by building and bushes. Static camera faces North.
6	Football field with ash surface build behind a supermarket next a road and crop-fields. Long side of the football field is surrounded by bushes. Static camera faces mostly North-West.
7	Backyard in an urban-village with both concrete and grass surface. Terrace has a garden table and chairs standing around. Static camera faces mostly West.
8	Parking lot next to allotments in a city-area. Static camera faces mostly North-East.
9	Meadow next to a building. Static camera faces South.
10	City park in a metropolitan area behind a city mall. Park is surrounded by buildings, a playing ground, football and basketball fields. Static camera faces mostly North.
11	City park in a metropolitan area. Static camera faces mostly South.

Table A.2.: Per-session meta-information. We provide the individual session count, duration of each session, number of activities performed during the session, location ID (LID) the session was performed at, approximate time of the year and day and weather conditions during recording time. More detailed information on each location can be found in Table A.1 using the location ID. * Additional recordings performed by sbj_0 in summer. † Additional recordings performed by sbj_14 in summer.

Participant	Session	Duration	# Activities	Month	Time-of-day	LID	Weather conditions
sbj_0	1	16:33:30	7	mid-Oct.	morning	1	sunny, $\approx 10^\circ\text{C}$
sbj_0	2	11:55:00	6	mid-Oct.	afternoon	1	partly-cloudy, $\approx 10^\circ\text{C}$
sbj_0	3	18:06:00	7	late-Oct.	afternoon	1	partly-cloudy, $\approx 20^\circ\text{C}$
sbj_1	1	20:20:00	9	late-Oct.	afternoon	1	sunny, $\approx 15^\circ\text{C}$
sbj_1	2	25:58:00	9	early-Nov.	afternoon	1	sunny, $\approx 10^\circ\text{C}$
sbj_2	1	32:24:00	9	early-Nov.	morning	1	sunny, $\approx 10^\circ\text{C}$
sbj_2	2	25:08:00	9	mid-Jan.	afternoon	2	cloudy, after rain, $\approx 0^\circ\text{C}$
sbj_2	3	01:52:00	1	mid-Feb.	afternoon	3	sunny, $\approx 5^\circ\text{C}$
sbj_3	1	33:34:00	10	mid-Nov.	afternoon	4	sunny, $\approx 5^\circ\text{C}$
sbj_3	2	25:52:00	6	mid-Nov.	afternoon	4	partly-cloudy, $\approx 10^\circ\text{C}$
sbj_3	3	06:24:00	2	mid-Nov.	afternoon	4	sunny, $\approx 10^\circ\text{C}$
sbj_3	4	03:41:00	2	late-Jan.	afternoon	5	cloudy, snowy, $\approx -5^\circ\text{C}$
sbj_4	1	24:07:30	9	mid-Nov.	midday	1	foggy, cloudy, windy, $\approx 5^\circ\text{C}$
sbj_4	2	29:04:00	9	late-Nov.	afternoon	1	partly-cloudy, $\approx 5^\circ\text{C}$
sbj_5	1	19:48:30	9	mid-Nov.	afternoon	1	sunny, $\approx 10^\circ\text{C}$
sbj_5	2	16:02:00	9	end-Nov.	afternoon	1	cloudy, $\approx 5^\circ\text{C}$
sbj_6	1	23:52:00	10	end-Nov.	afternoon	1	foggy, $\approx 5^\circ\text{C}$
sbj_6	2	17:51:30	8	end-Jan.	morning	5	cloudy, snowy, $\approx -5^\circ\text{C}$
sbj_7	1	22:48:00	9	late-Dec.	morning	6	partly-sunny, $\approx 10^\circ\text{C}$
sbj_7	2	24:45:00	9	late-Dec.	midday	6	partly-sunny, $\approx 10^\circ\text{C}$
sbj_8	1	20:00:00	9	late-Dec.	midday	6	partly-cloudy, $\approx 10^\circ\text{C}$
sbj_8	2	21:35:00	9	late-Jan.	afternoon	7	cloudy, $\approx 0^\circ\text{C}$
sbj_9	1	18:50:00	9	early-Jan.	afternoon	8	cloudy, $\approx 10^\circ\text{C}$
sbj_9	2	17:16:00	9	early-Jan.	afternoon	8	cloudy, $\approx 10^\circ\text{C}$
sbj_10	1	21:42:00	9	mid-Jan.	afternoon	5	rainy, windy, $\approx 5^\circ\text{C}$
sbj_10	2	21:04:00	9	early-Feb.	afternoon	5	rainy, windy, $\approx 5^\circ\text{C}$
sbj_10	3	23:39:00	9	mid-Feb.	afternoon	9, 3	sunny, cloudy, windy, $\approx 5^\circ\text{C}$
sbj_11	1	17:41:00	9	mid-Jan.	morning	5	cloudy, rainy, $\approx 5^\circ\text{C}$
sbj_11	2	19:21:00	9	mid-Jan.	midday	5	cloudy, rainy, $\approx 5^\circ\text{C}$
sbj_12	1	27:08:00	9	mid-Jan.	afternoon	5	cloudy, windy, $\approx 0^\circ\text{C}$
sbj_12	2	27:22:00	9	late-Feb.	afternoon	5	partly-sunny, windy, $\approx 0^\circ\text{C}$
sbj_13	1	30:08:00	9	mid-Jan.	afternoon	5, 3	sunny, $\approx 0^\circ\text{C}$
sbj_13	2	36:10:00	9	mid-Jan.	afternoon	5, 3	sunny, $\approx 0^\circ\text{C}$
sbj_14	1	22:18:00	9	mid-Jan.	afternoon	5, 3	sunny, $\approx -5^\circ\text{C}$
sbj_14	2	31:03:00	9	mid-Jan.	afternoon	5, 3	cloudy, $\approx -5^\circ\text{C}$
sbj_15	1	23:17:00	9	late-Jan.	afternoon	5, 3	cloudy, $\approx 0^\circ\text{C}$
sbj_15	2	20:06:00	9	late-Jan.	afternoon	5, 3	cloudy, $\approx 0^\circ\text{C}$
sbj_16	1	26:34:00	9	early-Feb.	midday	10	partly-sunny, $\approx 10^\circ\text{C}$
sbj_16	2	31:56:00	9	early-Feb.	midday	10	partly-sunny, $\approx 10^\circ\text{C}$
sbj_17	1	23:16:00	9	early-Feb.	afternoon	1	sunny, $\approx 0^\circ\text{C}$
sbj_17	2	28:15:00	9	early-Feb.	afternoon	3	sunny, $\approx 0^\circ\text{C}$
sbj_18*	1	19:15:00	9	mid-Aug.	afternoon	1	sunny, $\approx 25^\circ\text{C}$
sbj_18*	2	18:06:00	9	mid-Aug.	afternoon	1	sunny, $\approx 30^\circ\text{C}$
sbj_19†	1	25:48:00	9	mid-Aug.	afternoon	1	sunny, $\approx 25^\circ\text{C}$
sbj_19†	2	30:07:00	9	mid-Aug.	afternoon	1	sunny, $\approx 30^\circ\text{C}$
sbj_20	1	19:45:00	9	early-Mar.	afternoon	3	cloudy, $\approx 5^\circ\text{C}$
sbj_20	2	24:07:00	9	early-Apr.	afternoon	3	cloudy, $\approx 10^\circ\text{C}$
sbj_21	1	18:13:00	9	early-Mar.	midday	9	sunny, $\approx 15^\circ\text{C}$
sbj_21	2	16:56:00	9	early-Mar.	midday	9	sunny, $\approx 15^\circ\text{C}$
sbj_22	1	16:25:00	9	late-Mar.	midday	11	sunny, $\approx 10^\circ\text{C}$
sbj_22	2	15:23:00	9	late-Mar.	midday	11	sunny, $\approx 10^\circ\text{C}$
sbj_23	1	22:42:00	9	late-Mar.	midday	11	sunny, $\approx 10^\circ\text{C}$
sbj_23	2	19:54:00	9	late-Mar.	midday	11	sunny, $\approx 10^\circ\text{C}$

Table A.3.: Per subject answers to the questionnaire handed to participants after having completed all sessions. The questionnaire collected vital information (gender (G), left- or righthanded (L/R), age, height and weight) as well as workout-specific questions, i.e. frequency and type of private workouts and number of activities, part of the WEAR dataset, which were known in advance and regularly conducted in private workouts. Note that sbj_18 and sbj_19 are excluded from this list as they are identical to sbj_0 and sbj_14 respectively.

Subject	G	L/R	Age	Height	Weight	Private Workouts		Activities	
						Frequency	Type	Known	Regularly
sbj_0	M	R	≥ 40	180-189	70-79	5	Cycling	5	0
sbj_1	M	R	25-29	170-179	60-69	3	Hiking	11	0
sbj_2	M	R	25-29	180-189	80-89	5	Gym, Cardio	18	9
sbj_3	M	R	35-39	170-179	70-79	4-5	Gym, Basketball, Cardio	18	9
sbj_4	M	R	25-29	180-189	60-69	0	Table-tennis	18	0
sbj_5	F	R	30-34	160 -169	N/A	2-3	Freeletics	16	9
sbj_6	F	R	25-29	150-159	50-59	1	Gym	9	0
sbj_7	M	R	30-34	180-189	80-89	5	Gym, Cardio	18	5
sbj_8	F	R	25-29	170-179	60-69	2-3	Volleyball, Yoga	15	7
sbj_9	F	R	25-29	150-159	50-59	7	Gym, Bicycling, Cardio, Ballet	18	7
sbj_10	F	R	20-24	160-169	50-59	5	Gym, Dancing, Yoga	15	7
sbj_11	F	R	25-29	160-169	50-59	3	Volleyball, Cardio, Yoga	18	11
sbj_12	F	R	20-24	170-179	60-69	4	Gym	17	8
sbj_13	M	R	20-24	≥ 190	90-99	2	Gym, Cardio	16	8
sbj_14	M	R	30-34	170-179	80-89	0	N/A	11	2
sbj_15	F	L	25-29	180-189	60-69	8	Rowing, Gym, Cycling, Cardio	18	9
sbj_16	M	R	20-24	180-189	60-69	2-3	Gym	15	3
sbj_17	M	R	25-29	180-189	70-79	4	Badminton, Bouldering, Hiking	15	5
sbj_20	M	R	25-29	170-179	70-79	N/A	N/A	12	N/A
sbj_21	M	R	20-24	170-179	70-79	2-3	Bouldering, Cardio	12	2
sbj_22	M	R	25-29	170-179	70-79	4	Badminton, Team Sport, Hiking	18	15
sbj_23	F	R	25-29	170-179	60-69	3-4	Tennis, Gym	17	4

with the activities detailed in the study protocol. The workout-specific questions were:

1. How many workouts (longer than 15 min) do you usually do per week?
2. Which kind of workout do you usually do (cycling, team sport, gym, cardio, yoga etc.)?
3. How many activities that are part of the workout plan did you know in advance?
4. How many activities that are part of the workout plan do you perform regularly yourself as part of your own workouts?

Table A.3 shows the answers to the questionnaire items for each participant. Note that, to protect the privacy of our study participants, we only asked for age, height and weight in ranges instead of exact values, and always provided the option to not answer the questions if preferred.

Table A.4.: Test results of human activity recognition approaches based on body-worn IMU (Inertial), vision (Camera) and combined (I + C) features for different clip lengths (CL) on the test dataset mentioned in the main paper evaluated in terms of precision (P), recall (R), F1-score and mean average precision (mAP) for different temporal intersection over union (tIoU) thresholds. Best results per modality are in **bold**.

	Model	CL	P	R	F1	mAP					
						0.3	0.4	0.5	0.6	0.7	Avg
Inertial	Shallow D.	0.5s	88.47	86.02	85.55	75.86	74.24	73.06	70.78	68.10	72.41
	A-and-D	0.5s	85.42	79.24	79.67	65.36	63.51	59.64	56.73	56.01	60.25
	ActionFormer	0.5s	76.11	87.24	79.00	89.37	87.68	76.95	59.63	38.26	70.38
	TriDet	0.5s	81.48	86.63	82.00	90.26	88.02	80.21	68.33	58.38	77.04
	Shallow D.	1s	90.10	86.10	86.20	73.94	72.70	71.60	69.22	67.99	71.09
	A-and-D	1s	87.48	83.62	83.97	71.88	70.58	67.63	63.97	61.98	67.21
	ActionFormer	1s	78.69	86.72	80.52	91.68	90.48	89.23	87.23	83.17	88.36
	TriDet	1s	79.55	84.61	80.04	90.28	89.54	88.37	87.49	84.83	88.10
	Shallow D.	2s	88.06	84.73	84.38	69.31	67.73	65.61	63.55	60.85	65.41
	A-and-D	2s	88.67	85.36	85.43	72.81	70.47	67.39	64.90	62.59	67.63
	ActionFormer	2s	72.57	83.54	74.95	89.95	88.86	87.38	86.03	80.43	86.53
	TriDet	2s	76.49	83.24	77.25	89.85	89.17	87.97	85.37	80.77	86.63
Camera	ActionFormer	0.5s	65.58	76.04	67.32	83.72	82.85	78.84	58.43	37.52	68.27
	TriDet	0.5s	72.93	75.93	71.89	84.51	82.48	76.46	67.61	57.60	73.73
	ActionFormer	1s	70.70	78.17	71.90	86.09	86.09	85.98	85.49	81.56	85.04
	TriDet	1s	73.00	78.64	72.49	86.65	86.42	86.17	84.96	83.21	85.48
	ActionFormer	2s	68.97	75.79	68.39	86.12	85.71	85.21	84.00	79.11	84.03
	TriDet	2s	73.41	77.31	71.38	86.44	86.33	86.23	85.49	82.74	85.45
I + C	ActionFormer	0.5s	78.28	88.55	80.97	92.07	91.69	86.63	68.15	41.08	75.92
	TriDet	0.5s	86.00	89.12	86.23	92.18	90.15	83.11	75.24	64.59	81.06
	ActionFormer	1s	82.59	89.07	84.26	94.24	94.06	93.80	93.29	89.51	92.98
	TriDet	1s	85.72	90.78	86.98	94.47	94.29	93.77	92.69	90.39	93.12
	ActionFormer	2s	76.80	86.35	78.86	91.53	91.28	90.89	88.83	84.33	89.37
	TriDet	2s	80.94	87.19	82.19	93.43	92.81	92.01	91.34	89.63	91.84

A.3. Supplementary Experiments and Figures

A.3.1. Detailed Test Results

Table A.4 provides additional results on the test set for different window lengths (0.5, 1 and 2 seconds). We chose to use hyperparameters and postprocessing in accordance with the LOSO cross-validation experiments. That is, predictions of the inertial-based architectures were smoothed using a majority filter of 10 seconds and predictions of the Temporal Action Localization (TAL) models were score thresholded using a threshold of 0.1 based on the prediction confidence of the model associated with each segment.

A.3.2. Ablation Study on Postprocessing

The following details ablation experiments conducted to demonstrate the effectiveness and validity of the applied postprocessing described in the experiments section of the main paper. Figure A.3 illustrates the effect the majority vote filter has on the prediction stream of the inertial-based models. One can see that without applying a majority vote filter, inertial-based architectures produce a large amount of non-coherent segments. This is due to the fact that during training, inertial models are not explicitly trained to predict coherent segments, but rather predict a continuous stream of windowed data. The models therefore tend to show a lot of intermediate switches in-between activity labels which causes mAP scores of inertial-based architectures to be substantially lower than scores of vision-based models. We therefore make use of a majority vote filter to erase short activity-label switches. Table A.5 and A.6 shows experimental results of applying different-sized majority vote filters (5, 10, 15, 20 and 25 seconds) compared to applying no filter. Interestingly, results (see Table A.5 and A.6) not only demonstrate the effectiveness of the majority vote filter through a substantial increase in mAP scores, yet also show that said increase does not come at the cost of a decreased F1-score, but rather an increase. Table A.5 and A.6 further show a majority vote filter of 10 seconds being most effective resulting in the highest F1-score.

TAL models are not trained on an explicitly modelled NULL-class. This means, that unlike inertial models such as the DeepConvLSTM, TAL models are only able to predict segments with activity labels other than the NULL-class. With both models being set to predict up to 2000 action segments per video, the unprocessed prediction results resulted in activity streams such as illustrated in Figure A.4. One can see that almost all samples have been assigned an activity label, leaving only a few data to be predicted as NULL, ultimately resulting in a substantially lower NULL-class accuracy than compared to inertial-based models mentioned in this paper. We therefore increased the score-threshold of both the ActionFormer and TriDet model, eliminating low-scoring segments and replacing them with NULL (see Figure A.4). This improved classification performance of the ActionFormer (see Table A.7) and TriDet model (see Table A.8) significantly across all experiments (i.e. using inertial, vision and a combined setup as input data), while only marginally decreasing mAP scores. Table A.8 further shows 0.1 being the most effective identified threshold of our ablation study, resulting in the highest F1-score of the TAL models.

Table A.5.: Ablation experiments on the effect of different-sized majority vote (MV) filters (5, 10, 15, 20 and 25 seconds) on the raw prediction results (0 seconds) of the shallow DeepConvLSTM model. We report results on the three employed window sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Best results per clip-length are in **bold**.

	MV filter	P	R	F1	mAP					
					0.3	0.4	0.5	0.6	0.7	Avg
0.5 window	0 sec	68.7	68.42	65.96	5.62	4.53	3.55	3.01	2.66	3.87
	5 sec	75.23	74.69	72.05	50.99	48.52	46.70	44.54	42.22	46.59
	10 sec	74.81	74.93	72.26	60.03	57.91	55.54	54.19	52.31	55.99
	15 sec	74.38	74.85	72.24	62.59	61.03	58.82	56.71	54.59	58.75
	20 sec	74.19	74.55	71.96	64.35	62.85	60.19	57.97	55.34	60.14
	25 sec	73.77	74.08	71.53	65.09	63.58	60.96	58.91	56.39	60.99
1.0 window	0 sec	74.00	72.11	70.36	12.16	10.50	9.16	7.58	6.89	9.26
	5 sec	77.84	77.08	75.04	55.53	53.27	51.33	48.98	46.28	51.08
	10 sec	77.71	77.41	75.44	63.37	61.35	58.91	57.02	55.05	59.14
	15 sec	77.59	77.32	75.31	66.49	64.55	62.51	60.70	58.18	62.49
	20 sec	77.34	76.99	75.05	67.90	66.14	64.37	61.86	59.26	63.91
	25 sec	76.82	76.55	74.66	69.02	67.53	65.38	62.50	59.67	64.82
2.0 window	0 sec	75.85	73.61	72.08	23.54	20.86	17.93	16.04	14.37	18.55
	5 sec	78.18	76.72	75.06	56.09	53.73	51.33	49.26	46.41	51.36
	10 sec	78.00	77.19	75.43	65.14	63.58	60.98	59.15	57.39	61.25
	15 sec	77.60	77.08	75.25	67.10	65.89	63.24	61.27	59.24	63.35
	20 sec	77.39	76.88	75.02	67.74	66.34	64.15	62.25	60.10	64.11
	25 sec	76.22	71.39	70.71	64.05	61.10	58.24	55.78	52.66	58.37

Table A.6.: Ablation experiments on the effect of different-sized majority vote (MV) filters (5, 10, 15, 20 and 25 seconds) on the raw prediction results (0 seconds) of the improved Attend-and-Discriminate model. We report results on the three employed window sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Best results per clip length are in **bold**.

	MV filter	P	R	F1	mAP					
					0.3	0.4	0.5	0.6	0.7	Avg
0.5 window	0 sec	70.61	65.85	65.32	3.60	3.04	2.42	1.95	1.60	2.52
	5 sec	76.72	72.87	71.90	47.90	44.53	42.34	39.80	38.09	42.53
	10 sec	76.54	73.04	72.10	57.51	54.59	52.22	49.47	46.94	52.15
	15 sec	76.43	72.68	71.80	61.04	58.36	55.96	53.03	50.74	55.83
	20 sec	76.28	72.06	71.29	62.94	59.99	57.14	54.34	52.06	57.30
	25 sec	77.10	76.51	74.67	69.09	67.81	65.51	63.45	61.51	65.47
1.0 window	0 sec	73.54	69.83	68.95	9.18	7.66	5.97	5.26	4.56	6.53
	5 sec	80.04	75.77	74.60	51.35	48.40	45.63	43.09	40.22	45.74
	10 sec	79.97	75.92	74.67	61.72	58.77	56.29	53.83	51.59	56.44
	15 sec	79.01	75.68	74.41	63.54	61.64	59.26	57.27	54.63	59.27
	20 sec	78.33	75.18	73.96	65.15	62.76	60.87	58.42	55.62	60.56
	25 sec	77.90	74.61	73.51	65.88	64.21	61.72	58.75	56.70	61.45
2.0 window	0 sec	76.51	73.63	72.36	18.80	16.00	13.31	10.78	9.76	13.73
	5 sec	81.10	77.78	76.60	53.55	51.18	47.96	45.72	42.31	48.15
	10 sec	80.96	78.42	77.10	63.52	61.57	58.82	56.88	54.41	59.04
	15 sec	80.30	78.29	76.92	67.27	65.35	62.55	59.97	57.31	62.49
	20 sec	79.77	77.88	76.49	67.77	65.78	63.17	60.69	58.23	63.13
	25 sec	79.48	77.35	76.01	68.58	66.80	64.81	62.24	59.56	64.40

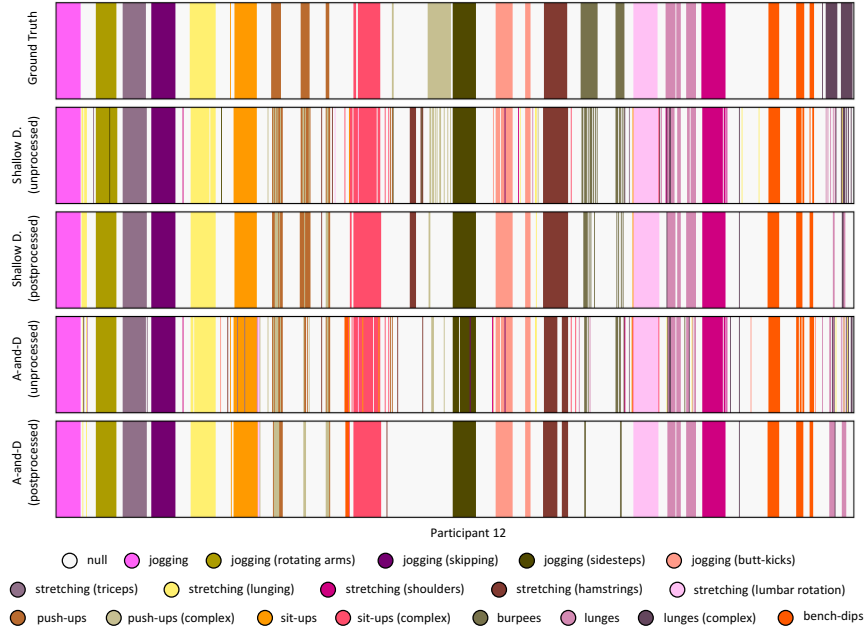


Figure A.3.: Color-coded comparison of the ground truth data (top row) with the raw and postprocessed (10 sec majority vote filter) activity streams of the shallow DeepConvLSTM and Attend-and-Discriminate (A-and-D) model. The illustrated activity stream is of a sample subject having trained using inertial data which is windowed using a 1 second sliding window with 50% overlap.

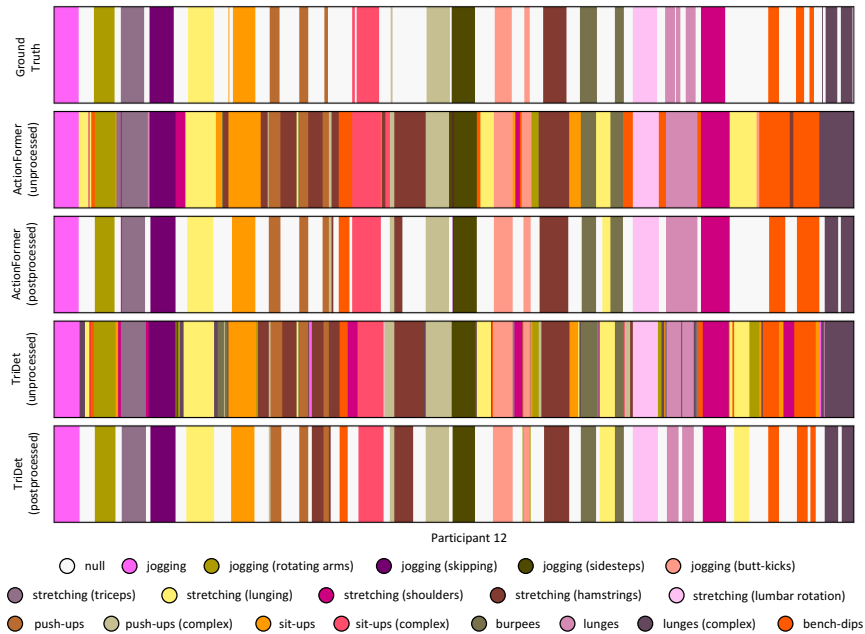


Figure A.4.: Color-coded comparison of the ground truth data (top row) with the raw and score-thresholded (0.1) activity streams of the TriDet model. The illustrated activity stream is of sample subject having trained the model using both inertial and vision data which is windowed using a 1 second sliding window with 50% overlap.

Table A.7.: ActionFormer score thresholding results ablation experiments on the effect of different score thresholds (0.05, 0.1, 0.15, 0.2 and 0.25) on the raw prediction results (0.0 threshold). We report results using three clip length window sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Best results per modality are in **bold**.

	Threshold	CL	P	R	F1	mAP						
						0.3	0.4	0.5	0.6	0.7	Avg	
Inertial	0.0	0.5s	52.51	77.38	59.62	85.50	81.74	72.97	58.06	41.71	67.99	
	0.05	0.5s	58.47	77.76	61.62	83.74	79.91	71.27	56.54	40.48	66.38	
	0.1	0.5s	63.09	78.17	66.80	81.12	77.47	69.08	53.98	38.56	64.04	
	0.15	0.5s	65.87	76.17	68.46	78.26	74.64	66.29	50.99	36.38	61.31	
	0.2	0.5s	66.00	71.29	66.72	73.47	70.02	61.97	47.03	34.04	57.30	
	0.25	0.5s	62.29	63.63	61.40	67.21	63.99	55.91	42.26	31.22	52.12	
	0.0	1.0s	55.94	77.49	59.74	85.90	83.55	80.83	75.59	66.74	78.52	
	0.05	1.0s	61.03	78.23	63.36	84.21	82.01	79.38	74.35	65.69	77.13	
	0.1	1.0s	65.88	78.44	68.40	81.91	79.75	76.99	72.09	64.33	75.01	
	0.15	1.0s	68.50	77.25	70.33	79.45	77.44	74.80	70.56	63.12	73.07	
	0.2	1.0s	69.24	74.71	70.16	76.61	74.59	72.16	68.38	61.47	70.64	
	0.25	1.0s	69.60	72.15	69.33	74.23	72.25	69.84	66.40	59.89	68.52	
	0.0	2.0s	52.26	76.33	57.19	83.96	81.04	78.12	73.70	66.10	76.58	
	0.05	2.0s	56.08	76.73	59.52	82.48	79.56	76.73	72.52	65.24	75.30	
	0.1	2.0s	61.56	76.63	64.57	80.14	77.24	74.41	70.24	63.18	73.04	
	0.15	2.0s	64.76	75.03	66.77	77.63	74.88	72.21	68.33	61.78	70.97	
	0.2	2.0s	67.22	72.67	67.67	75.11	72.54	70.10	66.38	60.14	68.86	
	0.25	2.0s	67.33	69.61	66.60	71.34	68.96	66.74	63.16	57.59	65.56	
	Camera	0.0	0.5s	48.38	72.36	54.31	85.47	82.48	76.21	61.63	45.21	70.20
		0.05	0.5s	57.02	73.50	59.01	84.92	81.93	75.69	60.82	44.10	69.49
		0.1	0.5s	60.86	72.98	62.54	81.07	78.48	72.13	57.34	41.27	66.06
		0.15	0.5s	61.74	70.15	62.56	75.94	73.45	67.82	53.54	38.37	61.83
		0.2	0.5s	60.12	65.24	60.16	69.51	67.27	61.75	48.47	35.46	56.49
		0.25	0.5s	55.92	58.25	54.97	60.86	58.95	54.24	43.59	32.19	49.97
0.0		1.0s	52.62	74.21	55.41	87.52	85.55	83.00	78.97	71.43	81.30	
0.05		1.0s	61.39	75.98	63.20	87.06	85.02	82.42	78.45	70.99	80.79	
0.1		1.0s	65.82	75.34	66.40	85.19	83.31	80.94	77.22	69.96	79.32	
0.15		1.0s	67.98	73.95	67.37	82.34	80.55	78.50	74.94	68.11	76.89	
0.2		1.0s	67.65	71.81	66.57	78.58	77.22	75.30	71.95	65.29	73.67	
0.25		1.0s	66.56	69.73	65.43	74.34	73.06	71.16	68.12	61.94	69.72	
0.0		2.0s	54.14	75.41	57.04	86.92	84.80	82.54	79.42	73.81	81.50	
0.05		2.0s	58.55	76.22	60.61	86.51	84.41	82.13	79.00	73.43	81.10	
0.1		2.0s	63.40	76.25	65.39	84.93	82.95	80.58	77.55	72.07	79.62	
0.15		2.0s	66.78	74.93	67.40	82.02	80.44	78.32	75.25	70.10	77.23	
0.2		2.0s	67.88	73.14	67.61	78.73	77.28	75.39	72.31	67.55	74.25	
0.25		2.0s	67.07	70.94	66.57	74.46	73.23	71.48	68.50	64.08	70.35	
Inertial + Camera		0.0	0.5s	57.97	82.71	65.13	91.13	88.98	82.88	69.44	48.25	76.13
		0.05	0.5s	67.05	83.92	70.09	89.42	87.24	81.22	67.80	46.88	74.51
		0.1	0.5s	71.45	83.77	74.51	86.04	83.96	77.80	64.41	43.76	71.19
		0.15	0.5s	72.26	80.29	74.26	81.37	79.66	73.87	60.31	40.82	67.21
		0.2	0.5s	70.39	74.64	71.03	76.21	74.66	69.39	56.22	37.89	62.87
		0.25	0.5s	66.93	67.82	66.00	70.10	68.55	63.61	50.82	35.08	57.63
	0.0	1.0s	59.04	81.82	63.89	90.25	88.65	85.98	81.92	75.17	84.39	
	0.05	1.0s	68.32	83.47	71.39	88.59	86.98	84.28	80.27	73.93	82.81	
	0.1	1.0s	72.87	83.24	75.26	86.82	85.27	82.46	78.26	72.33	81.03	
	0.15	1.0s	74.92	81.73	76.12	84.44	82.88	80.10	76.33	70.89	78.93	
	0.2	1.0s	74.98	79.55	75.56	81.76	80.31	77.85	74.31	69.39	76.73	
	0.25	1.0s	74.29	77.21	74.24	79.00	77.56	75.38	72.35	67.89	74.43	
	0.0	2.0s	55.98	79.41	60.41	87.78	85.31	81.75	77.93	71.45	80.84	
	0.05	2.0s	60.61	80.04	63.81	86.39	83.92	80.41	76.63	70.30	79.53	
	0.1	2.0s	65.86	80.17	68.82	84.18	81.66	78.17	74.59	68.74	77.47	
	0.15	2.0s	68.97	78.69	70.93	81.98	79.47	76.33	72.84	67.39	75.60	
	0.2	2.0s	71.12	76.70	71.74	79.34	77.04	73.97	70.79	65.66	73.36	
	0.25	2.0s	72.14	74.44	71.43	76.58	74.21	71.41	68.42	63.53	70.83	

Table A.8.: TriDet score thresholding results ablation experiments on the effect of different score thresholds (0.05, 0.1, 0.15, 0.2 and 0.25) on the raw prediction results (0.0 threshold). We report results using three clip length sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Best results per modality are in **bold**.

	Threshold	CL	P	R	F1	mAP					
						0.3	0.4	0.5	0.6	0.7	Avg
Inertial	0.0	0.5s	54.18	77.70	60.50	83.96	80.03	73.27	63.93	53.05	70.85
	0.05	0.5s	62.04	78.53	64.27	82.20	78.38	72.16	62.94	52.06	69.55
	0.1	0.5s	68.58	78.30	70.41	79.66	76.07	69.78	60.57	49.95	67.21
	0.15	0.5s	69.86	74.77	70.41	75.63	72.10	66.24	57.34	47.48	63.76
	0.2	0.5s	69.84	69.83	68.18	71.70	67.99	62.28	54.17	44.77	60.18
	0.25	0.5s	65.54	62.54	62.40	65.46	62.05	56.51	48.87	40.75	54.73
	0.0	1.0s	52.39	77.82	59.66	85.73	84.20	82.08	77.27	70.71	80.00
	0.05	1.0s	60.94	78.74	64.08	83.90	82.43	80.42	76.13	70.06	78.59
	0.1	1.0s	68.18	78.58	70.36	81.89	80.46	78.47	74.59	68.91	76.86
	0.15	1.0s	71.92	76.62	72.17	79.80	78.36	76.42	72.58	67.29	74.89
	0.2	1.0s	72.22	73.95	71.52	76.61	75.31	73.60	70.11	65.20	72.17
	0.25	1.0s	71.83	70.79	69.94	73.17	71.93	70.29	67.19	62.71	69.06
	0.0	2.0s	49.96	76.24	57.44	83.66	81.72	78.63	75.15	69.00	77.63
	0.05	2.0s	57.17	76.55	60.68	81.23	79.35	76.36	72.88	66.92	75.35
	0.1	2.0s	63.14	75.64	65.94	78.69	77.02	73.89	70.68	65.31	73.12
Camera	0.15	2.0s	66.58	72.86	67.38	75.45	73.97	70.93	68.13	63.23	70.34
	0.2	2.0s	67.75	70.27	67.12	72.66	71.22	68.27	65.86	61.48	67.90
	0.25	2.0s	66.75	67.33	65.61	68.75	67.40	64.53	62.50	58.67	64.37
	0.0	0.5s	48.68	72.06	54.16	85.34	81.82	77.26	68.62	59.33	74.47
	0.05	0.5s	58.86	73.47	59.99	84.71	81.15	76.52	67.78	58.51	73.73
	0.1	0.5s	64.69	72.89	64.84	80.46	76.95	72.62	64.46	55.52	70.00
	0.15	0.5s	65.10	68.81	63.72	74.67	71.32	67.34	59.43	51.23	64.80
	0.2	0.5s	62.64	63.16	60.19	67.93	64.64	61.17	53.63	46.58	58.79
	0.25	0.5s	57.95	57.43	55.79	59.97	57.09	53.79	47.40	41.57	51.96
	0.0	1.0s	49.99	73.89	55.56	87.72	86.46	84.83	82.18	78.56	83.95
	0.05	1.0s	61.68	75.51	62.67	87.11	85.80	84.14	81.43	77.87	83.27
	0.1	1.0s	67.42	75.21	66.88	84.88	83.49	82.12	79.75	76.24	81.30
	0.15	1.0s	70.30	73.96	68.32	81.95	80.62	79.36	77.23	74.00	78.63
	0.2	1.0s	70.76	71.98	67.95	78.68	77.46	76.21	74.22	71.38	75.59
	0.25	1.0s	69.58	69.64	66.67	74.53	73.54	72.40	70.70	68.33	71.90
Inertial + Camera	0.0	2.0s	50.21	74.28	56.32	85.93	84.86	82.71	80.34	77.16	82.20
	0.05	2.0s	59.34	75.22	60.83	85.25	84.21	82.09	79.75	76.59	81.58
	0.1	2.0s	65.53	75.36	65.91	82.94	82.00	80.22	78.13	75.20	79.70
	0.15	2.0s	68.58	74.02	67.66	79.67	78.85	77.10	75.28	72.66	76.71
	0.2	2.0s	69.12	72.05	67.61	76.02	75.27	73.60	71.95	69.39	73.25
	0.25	2.0s	67.22	69.05	65.74	71.38	70.70	69.42	68.16	65.70	69.07
	0.0	0.5s	59.15	82.23	65.24	90.07	87.62	82.36	74.05	61.73	79.17
	0.05	0.5s	69.69	83.84	72.14	87.86	85.55	80.58	72.50	60.50	77.40
	0.1	0.5s	75.74	82.79	76.83	84.45	82.07	77.12	68.81	57.16	73.92
	0.15	0.5s	76.74	79.06	76.20	80.72	78.37	73.26	65.01	54.01	70.28
	0.2	0.5s	74.38	73.15	72.15	75.13	73.42	68.33	60.81	50.38	65.61
	0.25	0.5s	68.44	64.87	65.10	67.25	65.71	61.35	54.47	45.76	58.91
	0.0	1.0s	55.95	80.96	62.89	90.17	89.16	87.12	84.07	79.12	85.93
	0.05	1.0s	67.87	82.58	70.67	88.63	87.63	85.55	82.44	77.64	84.38
	0.1	1.0s	73.39	82.00	75.09	86.42	85.48	83.35	80.39	75.64	82.26
	0.15	1.0s	75.96	80.49	76.38	83.94	83.07	80.87	78.01	73.62	79.90
	0.2	1.0s	76.64	78.38	75.95	80.89	80.09	78.20	75.57	71.80	77.31
	0.25	1.0s	75.76	76.03	74.64	77.79	77.09	75.29	73.02	69.64	74.56
	0.0	2.0s	53.20	80.55	61.50	88.39	86.47	84.31	81.56	76.90	83.52
	0.05	2.0s	62.39	81.35	66.42	86.78	84.86	82.72	80.09	75.59	82.01
	0.1	2.0s	69.19	81.09	71.98	84.66	82.79	80.66	78.11	74.15	80.07
	0.15	2.0s	72.36	78.89	73.41	81.72	79.92	77.93	75.41	71.94	77.38
	0.2	2.0s	72.78	75.99	72.69	78.24	76.71	74.79	72.53	69.45	74.34
	0.25	2.0s	72.94	74.01	71.94	75.85	74.47	72.71	70.58	67.84	72.29

Table A.9.: Results using only inertial sensors placed on the right wrist (RW) and right wrist + ankle (RW + RA) compared with using all sensors for different clip lengths (CL) on our WEAR dataset evaluated in terms of F1-score and mean average precision (mAP) averaged across different temporal intersection over union (tIoU) thresholds (0.3:0.7:0.1). Using only wrist-worn data can see a clear overall decrease across all evaluation metrics. Best results per modality are in **bold**.

	Model	CL	RW		RW + RA		all	
			F1	mAP	F1	mAP	F1	mAP
Inertial	Shallow D.	0.5s	60.50	27.56	72.42	47.26	72.26	55.99
	A-and-D	0.5s	65.65	32.38	69.60	45.66	72.10	52.15
	ActionFormer	0.5s	59.64	61.87	70.67	66.22	66.80	64.04
	TriDet	0.5s	62.05	64.64	74.06	71.02	70.41	67.21
	Shallow D.	1s	61.97	29.56	73.54	49.49	75.44	59.14
	A-and-D	1s	66.64	33.92	72.14	46.20	74.67	56.44
	ActionFormer	1s	61.62	72.80	73.13	77.88	68.40	75.01
	TriDet	1s	63.62	73.80	74.91	81.01	70.36	76.86
	Shallow D.	2s	60.74	29.26	74.48	50.72	75.43	61.25
	A-and-D	2s	66.19	34.50	73.98	51.36	77.10	59.04
	ActionFormer	2s	58.38	68.15	68.06	75.90	64.57	73.04
	TriDet	2s	60.48	68.22	70.90	76.81	65.94	73.12
I + C	ActionFormer	0.5s	71.75	70.48	76.54	72.80	74.51	71.19
	TriDet	0.5s	74.37	74.91	79.36	77.45	76.83	73.92
	ActionFormer	1s	73.49	79.46	78.18	83.55	75.26	81.03
	TriDet	1s	74.64	82.26	79.34	85.02	75.09	82.26
	ActionFormer	2s	69.21	79.69	73.37	81.13	68.82	77.47
	TriDet	2s	70.68	80.14	75.30	83.33	71.98	80.07

A.3.3. Ablation Study on the Selection of Inertial Sensors

Adding to results reported in the main paper, Table A.9 provides extended results on the influence of using only a subset of the inertial sensors, specifically using only (1) acceleration recorded from the right wrist and (2) acceleration recorded from both the right wrist and right ankle. Results show that using only acceleration data obtained from the right wrist significantly decreases predictive performance across all algorithms and metrics. Moreover, the value of additionally measuring acceleration at the ankles of participants is clearly underlined, as results again significantly increase, mostly on par compared to using all four inertial sensor locations. Interestingly, unlike the inertial-based architectures, results of the TAL models improve when excluding data captured by the left wrist and left ankle inertial sensors, which could be due to the dataset being biased towards right-handed participants (see Table A.3) and dominant hand movement being overall more consistent.

A.3.4. Ablation Study on Second Execution of Workout Sessions

Table A.10 provides additional results which compare repeated sessions for participants sbj_0 and sbj_14. The two participants were invited to perform the recording plan a second time. While one can see that improved results regarding sbj_0, suggesting potential learning effects of the correct execution of activities, this trend does not apply to sbj_14. Note that weather conditions (temperature and sunlight) significantly differ amongst the recordings – winter (first recording) compared to summer (2nd recording). Unlike our prior experiments, each algorithm is trained using the data of all but the validation subjects’ recordings, ensuring the validation subjects (sbj_0 and sbj_14) remain unseen during the training of each algorithm.

provides additional results which compare validation results obtained on the original, first recording of sbj_0 and sbj_14 with their second execution of the workout plan for different window lengths.

A.3.5. Ablation Study on Influence of Frequency of Inputs

With the frequencies both the camera (60 FPS) and inertial sensors (50 HZ) being set fairly high, the WEAR dataset allows to explore lower frequency experiments and their effect fewer datapoints per second might have on the predictive quality of the trained models. Table A.11 summarizes experiments conducted using only 50% and 20% of the available frequency for both types of sensors. Note that a clip length of 0.5 seconds was not explored during experiments as it was not possible anymore to extract two-stream I3D feature embeddings [27] as the amount of frames was lower than the required minimum input frames. Looking at results presented in Table A.11 one can see that all models trained using only inertial data suffered from lower frequency inputs with both classification and mAP scores decreasing. Contrarily, models trained using camera-based improved when using features extracted from videos with a lower FPS, which might be caused by Kinetics-400 [90], which was used for pretraining the I3D extraction method, on consisting of videos with a lower FPS than the WEAR dataset.

A.3.6. Attend-and-Discriminate Improvements

Instead of employing a plain Attend-and-Discriminate model as proposed by Abedin et al. [1], we incorporate architecture improvements suggested by Bock et al. [17]. Said architecture improvements are (1) using one instead of two recurrent layers, (2) increasing the amount of hidden units in the recurrent layer from 128 to 1024 and (3) scaling the convolutional kernel by the same factor the window size increases or decreases. Table A.12 shows performance difference gained from employing the improved Attend-and-Discriminate architecture by

Table A.10.: Comparison of obtained results of repeated sessions for participants sbj_0 and sbj_14 for different clip lengths (CL) on our WEAR dataset evaluated in terms of F1-score and mean average precision (mAP). These figures are, as in the earlier results, averaged across 3 runs using 3 different random seeds. For the first recording, both subjects' best results per modality are in underlined. For the second recording, both subjects' best results per modality are in **bold**. Unlike our prior experiments, each algorithm is trained using the data of all but the validation subjects' recordings, ensuring the validation subjects (sbj_0 and sbj_14) remain unseen during the training of each algorithm. All results are postprocessed as reported in the main paper.

	Model	CL	sbj_0				sbj_14			
			1st Recording		2nd Recording		1st Recording		2nd Recording	
			F1	mAP	F1	mAP	F1	mAP	F1	mAP
Inertial	Shallow D.	0.5s	71.82	42.54	89.09	78.46	85.14	64.03	81.92	50.32
	A-and-D	0.5s	72.35	36.98	86.34	67.69	78.36	48.64	73.14	43.13
	ActionFormer	0.5s	78.36	72.34	81.82	65.79	74.24	77.74	68.47	71.76
	TriDet	0.5s	80.36	69.34	84.73	76.12	78.99	87.08	73.86	80.56
	Shallow D.	1s	77.20	47.43	88.94	79.88	85.45	65.96	81.97	54.89
	A-and-D	1s	75.63	41.19	87.02	71.75	83.32	59.02	79.82	51.98
	ActionFormer	1s	75.48	72.70	78.83	87.22	76.36	92.02	71.29	84.42
	TriDet	1s	77.67	78.87	79.31	89.44	79.90	93.18	71.27	84.75
	Shallow D.	2s	76.57	46.99	87.57	79.69	86.37	71.02	84.88	59.60
	A-and-D	2s	68.48	39.71	87.49	77.10	85.75	65.61	85.48	51.92
	ActionFormer	2s	66.28	65.83	71.15	80.38	70.47	89.54	63.12	79.97
	TriDet	2s	71.27	64.33	72.94	86.30	75.54	91.81	65.01	88.61
C	ActionFormer	0.5s	62.17	57.52	75.29	68.31	63.06	65.84	69.57	77.35
	TriDet	0.5s	60.67	64.37	78.28	69.20	62.12	65.95	76.13	83.55
	ActionFormer	1s	62.97	63.64	73.86	88.70	58.67	78.13	71.27	89.42
	TriDet	1s	64.71	67.99	76.04	87.78	64.26	80.25	72.81	88.30
	ActionFormer	2s	54.72	55.36	75.03	87.78	56.66	81.11	70.52	91.94
	TriDet	2s	53.03	58.03	72.97	86.56	63.05	80.73	67.39	91.43
I + C	ActionFormer	0.5s	84.38	74.86	85.02	74.69	78.19	81.74	73.28	76.28
	TriDet	0.5s	84.07	66.72	82.42	74.63	82.46	87.74	78.34	80.04
	ActionFormer	1s	84.78	79.79	82.52	94.07	79.05	91.91	73.38	87.82
	TriDet	1s	83.03	82.55	83.54	93.18	78.35	97.66	70.78	86.76
	ActionFormer	2s	65.06	74.76	79.31	87.59	74.81	93.57	66.96	87.44
	TriDet	2s	76.16	77.61	76.34	88.40	76.89	96.23	68.73	84.22

Table A.11.: Results of evaluating different frequencies (Freq.) as input for different clip lengths (CL) on our WEAR dataset. Features were downsampled to be only 50% (30 FPS and 25 Hz) and 20% (12 FPS and 10 Hz) of the original frequency input. While predictive performance of inertial models decreases with a lower frequency input, camera and combined models increase in performance when lower frequency inputs. Best results per modality are in **bold**.

	Threshold	Freq.	CL	P	R	F1	mAP					
							0.3	0.4	0.5	0.6	0.7	Avg
Inertial	Shallow D.	Orig	1s	77.71	77.41	75.44	63.37	61.35	58.91	57.02	55.05	59.14
	Shallow D.	50%	1s	75.84	76.37	73.60	61.08	58.90	57.02	55.21	52.97	57.04
	Shallow D.	20%	1s	76.02	76.78	73.89	61.61	59.77	57.57	55.88	53.75	57.72
	A-and-D	Orig	1s	79.97	75.92	74.67	61.72	58.77	56.29	53.83	51.59	56.44
	A-and-D	50%	1s	79.00	75.33	74.35	59.69	57.12	54.46	51.98	48.96	54.44
	A-and-D	20%	1s	77.53	79.12	76.00	62.55	60.65	58.75	56.94	54.51	58.68
	ActionFormer	Orig	1s	65.88	78.44	68.40	81.91	79.75	76.99	72.09	64.33	75.01
	ActionFormer	50%	1s	66.09	77.89	68.24	81.74	79.30	75.85	69.88	62.68	73.89
	ActionFormer	20%	1s	64.43	77.30	67.22	81.39	79.33	76.25	71.49	63.39	74.37
	TriDet	Orig	1s	68.18	78.58	70.36	81.89	80.46	78.47	74.59	68.91	76.86
	TriDet	50%	1s	67.47	77.09	69.07	81.59	80.02	77.89	73.83	67.38	76.14
	TriDet	20%	1s	66.80	76.46	68.50	80.74	79.32	77.19	73.04	65.89	75.24
	Shallow D.	Orig	2s	78.00	77.19	75.43	65.14	63.58	60.98	59.15	57.39	61.25
	Shallow D.	50%	2s	78.60	75.23	74.07	60.99	57.86	54.78	52.53	50.18	55.27
	Shallow D.	20%	2s	75.02	76.76	73.38	60.42	58.43	55.87	54.11	52.13	56.19
	A-and-D	Orig	2s	80.96	78.42	77.10	63.52	61.57	58.82	56.88	54.41	59.04
	A-and-D	50%	2s	78.60	75.23	74.07	60.99	57.86	54.78	52.53	50.18	55.27
	A-and-D	20%	2s	79.25	76.13	74.76	61.43	59.55	56.85	54.80	52.55	57.04
	ActionFormer	Orig	2s	61.56	76.63	64.57	80.14	77.24	74.41	70.24	63.18	73.04
	ActionFormer	50%	2s	61.84	76.28	64.67	79.30	76.10	73.53	69.20	62.58	72.14
	ActionFormer	20%	2s	60.40	74.65	62.99	78.29	74.95	71.67	67.22	59.82	70.39
	TriDet	Orig	2s	63.14	75.64	65.94	78.69	77.02	73.89	70.68	65.31	73.12
	TriDet	50%	2s	63.22	74.52	65.52	77.99	76.18	73.49	69.94	63.06	72.13
	TriDet	20%	2s	63.01	73.43	64.84	77.21	75.22	72.71	69.75	62.99	71.58
	ActionFormer	Orig	1s	65.82	75.34	66.40	85.19	83.31	80.94	77.22	69.96	79.32
	ActionFormer	50%	1s	69.37	78.23	70.09	87.20	85.40	83.05	79.54	72.58	81.55
	ActionFormer	20%	1s	67.40	77.26	68.73	86.03	84.37	82.55	78.23	70.94	80.43
	TriDet	Orig	1s	67.42	75.21	66.88	84.88	83.49	82.12	79.75	76.24	81.30
	TriDet	50%	1s	70.92	77.88	70.19	86.62	85.22	83.42	80.75	77.10	82.62
	TriDet	20%	1s	69.52	76.68	68.71	84.58	83.49	82.12	80.12	75.14	81.09
Camera	ActionFormer	Orig	2s	63.40	76.25	65.39	84.93	82.95	80.58	77.55	72.07	79.62
	ActionFormer	50%	2s	67.55	79.65	69.36	86.99	85.34	83.10	80.54	75.00	82.19
	ActionFormer	20%	2s	65.70	76.30	66.59	84.90	83.02	81.09	77.77	71.33	79.62
	TriDet	Orig	2s	65.53	75.36	65.91	82.94	82.00	80.22	78.13	75.20	79.70
	TriDet	50%	2s	69.19	78.70	69.81	86.31	85.30	83.64	80.99	76.86	82.62
	TriDet	20%	2s	68.36	76.60	68.28	84.02	82.48	80.62	78.08	73.92	79.82
	ActionFormer	Orig	1s	72.87	83.24	75.26	86.82	85.27	82.46	78.26	72.33	81.03
	ActionFormer	50%	1s	74.60	84.85	77.19	87.88	86.87	84.34	79.84	73.00	82.39
	ActionFormer	20%	1s	74.73	84.27	76.79	87.99	86.63	83.78	80.32	72.86	82.31
	TriDet	Orig	1s	73.39	82.00	75.09	86.42	85.48	83.35	80.39	75.64	82.26
	TriDet	50%	1s	76.60	84.44	78.00	87.80	86.87	85.43	82.37	76.98	83.89
	TriDet	20%	1s	76.66	83.32	77.00	88.59	87.69	86.16	83.63	78.53	84.92
Inertial + Camera	ActionFormer	Orig	2s	65.86	80.17	68.82	84.18	81.66	78.17	74.59	68.74	77.47
	ActionFormer	50%	2s	68.74	82.46	71.65	86.73	84.12	80.99	77.49	71.53	80.17
	ActionFormer	20%	2s	69.56	82.26	71.97	86.58	83.99	80.78	76.56	69.49	79.48
	TriDet	Orig	2s	69.19	81.09	71.98	84.66	82.79	80.66	78.11	74.15	80.07
	TriDet	50%	2s	69.84	80.94	72.17	84.73	83.09	80.22	78.31	74.06	80.08
	TriDet	20%	2s	71.84	82.26	73.46	86.01	84.11	81.92	79.78	75.73	81.51

Table A.12.: Results demonstrating the effectiveness of made modifications to the Attend-and-Discriminate model. We compare the plain original model with an optimised version (1-layered LSTM with 1024 hidden units and an adjusted convolutional kernel sizes). We report results on the three employed window sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Note that results are reported with no postprocessing applied.

	Model	P	R	F1	mAP					
					0.3	0.4	0.5	0.6	0.7	Avg
0.5s	Original A-and-D	65.72	71.04	65.91	4.98	4.13	3.16	2.72	2.24	3.45
	Optimised A-and-D	70.61	65.85	65.32	3.60	3.04	2.42	1.95	1.60	2.52
1.0s	Original A-and-D	66.04	71.72	66.21	8.02	6.84	5.28	4.69	4.12	5.79
	Optimised A-and-D	73.54	69.83	68.95	9.18	7.66	5.97	5.26	4.56	6.53
2.0s	Original A-and-D	68.29	74.05	68.42	16.54	14.33	12.23	10.91	9.74	12.75
	Optimised A-and-D	76.51	73.63	72.36	18.80	16.00	13.31	10.78	9.76	13.73

comparing it to the original architecture. Note that results are reported without having applied any postprocessing.

A.3.7. Longer vs. Shorter Training Runs

As mentioned in the main paper, all inertial-based architectures are trained for 100 epochs as compared to 30 epochs. To compensate for longer training times we employ a step-wise learning rate schedule with a step size of 10 epochs and a decay rate of 0.9 [1]. Table A.13 shows results of the longer training runs remain stable compared to a shorter training time of 30 epochs.

A.3.8. Additional Visualizations of LOSO Experiments

In addition to the visualisations supplied in the main paper, the following provides supplementary visualizations of the LOSO cross-validation experiments for further analysis. All models mentioned in this section were trained using a clip length of 1.0 second with a 50% overlap. Predictions made by the TAL models were filtered using a score threshold of 0.1 and predictions made by inertial-based architectures were filtered using a majority vote filter of 10 seconds. Figure A.5 provides confusion matrices of the ActionFormer [204] being applied using inertial, camera and combined (inertial + camera) features. Figure A.6 provides Confusion matrices of the shallow DeepConvLSTM [17] and improved Attend-and-Discriminate [1] applied on inertial data. Figure A.8 shows a color-coded visualisation of predictions streams of all models mentioned in the results table of the main paper. Figure A.7 delivers a side-by-side comparison of the confusion matrices of prediction streams of all models in-

Table A.13.: Results demonstrating the effectiveness of longer training times on the inertial-based models. Compared are the shallow DeepConvLSTM and improved Attend-and-Discriminate model using either a short training time (30 epochs and no step-wise learning rate schedule (LRS)) or long training time (100 epochs and LRS). We report results on the three employed window sizes (0.5, 1.0 and 2.0 seconds) each with a 50% overlap. Note that results are reported with no postprocessing applied.

	Model	Epochs	LRS	P	R	F1	mAP					
							0.3	0.4	0.5	0.6	0.7	Avg
0.5s	Shallow D.	30		65.18	71.14	65.49	5.73	4.79	3.97	3.45	3.06	4.20
	Shallow D.	100	✓	68.70	68.42	65.96	5.62	4.53	3.55	3.01	2.66	3.87
	A-and-D	30		67.36	68.64	65.48	4.10	3.12	2.57	2.13	1.87	2.76
	A-and-D	100	✓	70.61	65.85	65.32	3.60	3.04	2.42	1.95	1.60	2.52
1.0s	Shallow D.	30		68.68	74.60	68.92	12.39	10.83	9.37	8.06	7.24	9.58
	Shallow D.	100	✓	73.54	69.83	68.95	9.18	7.66	5.97	5.26	4.56	6.53
	A-and-D	30		70.49	72.15	68.74	9.38	7.86	6.18	5.26	4.67	6.67
	A-and-D	100	✓	73.54	69.83	68.95	9.18	7.66	5.97	5.26	4.56	6.53
2.0s	Shallow D.	30		71.45	77.04	71.55	22.54	20.30	18.11	16.42	14.84	18.44
	Shallow D.	100	✓	75.85	73.61	72.08	23.54	20.86	17.93	16.04	14.37	18.55
	A-and-D	30		73.58	75.94	72.28	18.88	16.35	13.72	11.91	10.65	14.30
	A-and-D	100	✓	76.51	73.63	72.36	18.80	16.00	13.31	10.78	9.76	13.73

volved in the *Oracle*-late-fusion-approach analysis mentioned in the main paper. Figure A.7 shows that a joint learning of both modalities particularly improves differentiation between the NULL-class and the activity classes resulting in better action boundaries, i.e. higher mAP scores, and classification scores.

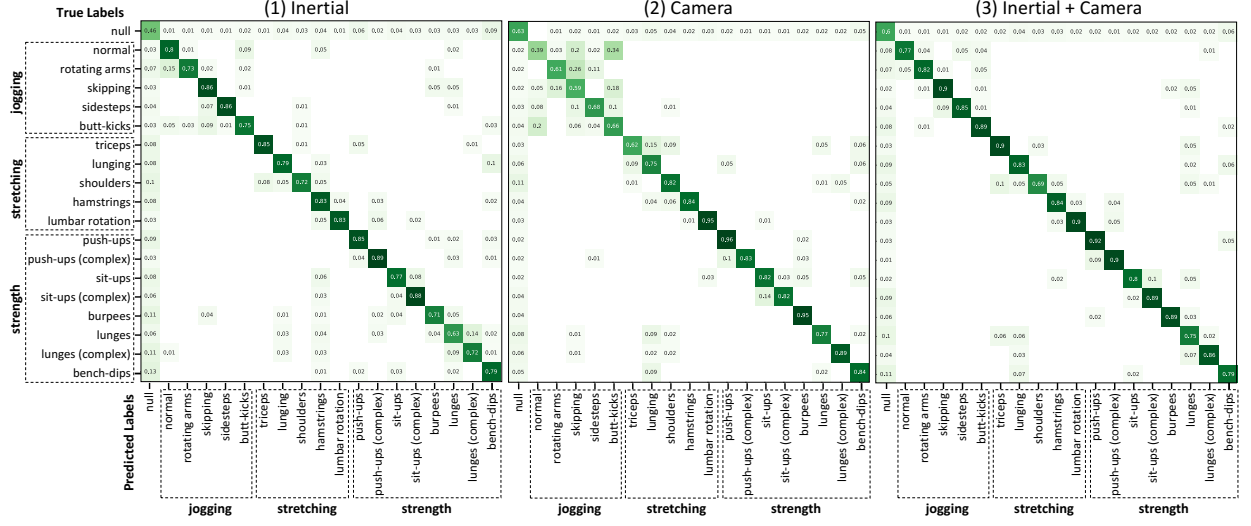


Figure A.5.: Confusion matrices of the ActionFormer being applied using only inertial, vision (camera) and both combined (inertial + camera).

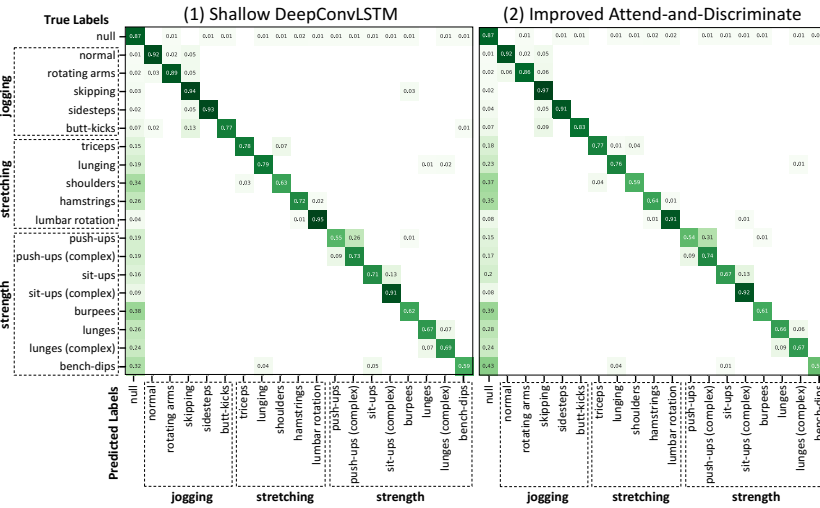


Figure A.6.: Confusion matrices of the shallow DeepConvLSTM and improved Attend-and-Discriminate.

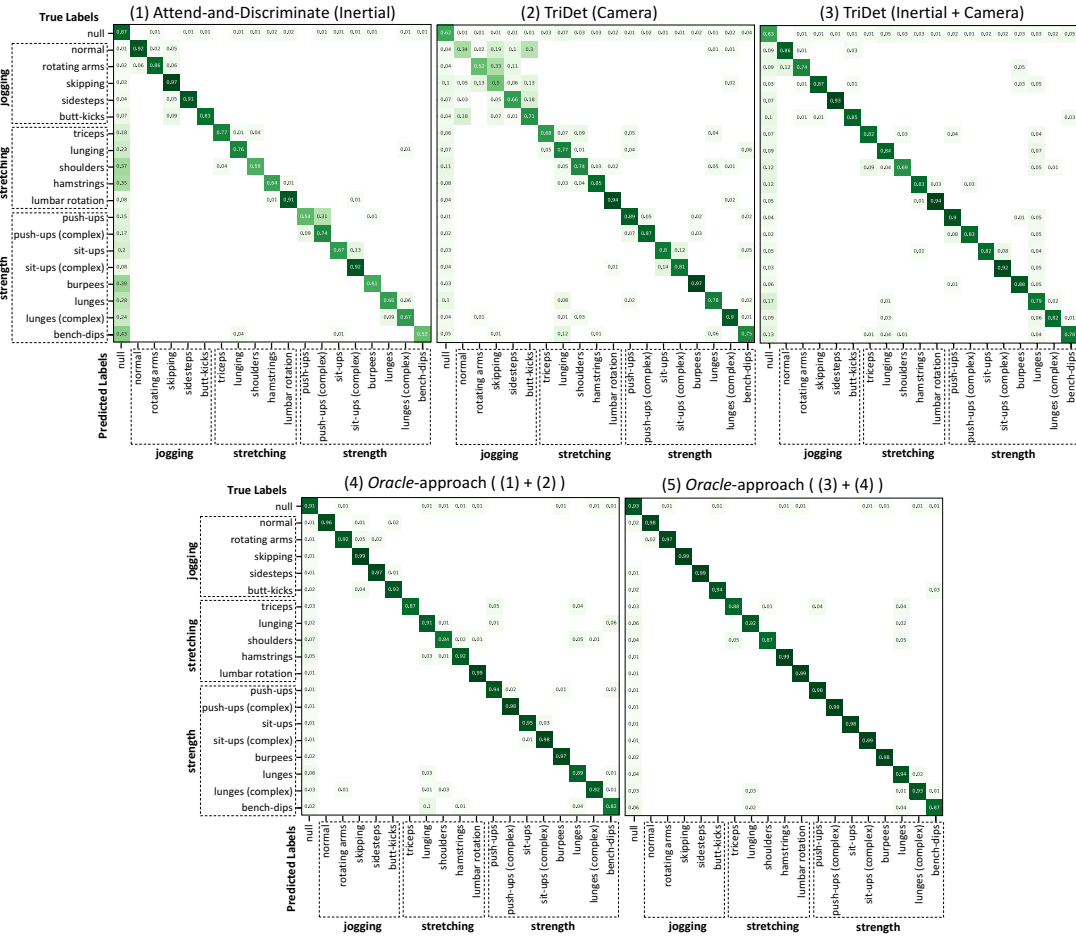


Figure A.7.: Confusion matrices of the best (1) inertial model (Attend-and-Discriminate), (2) vision model (TriDet) and (3) combined model (vision + inertial) compared with (4) an *Oracle*-combination of the inertial and camera as well (5) *Oracle*-combination of the previous oracle with the combined approach.

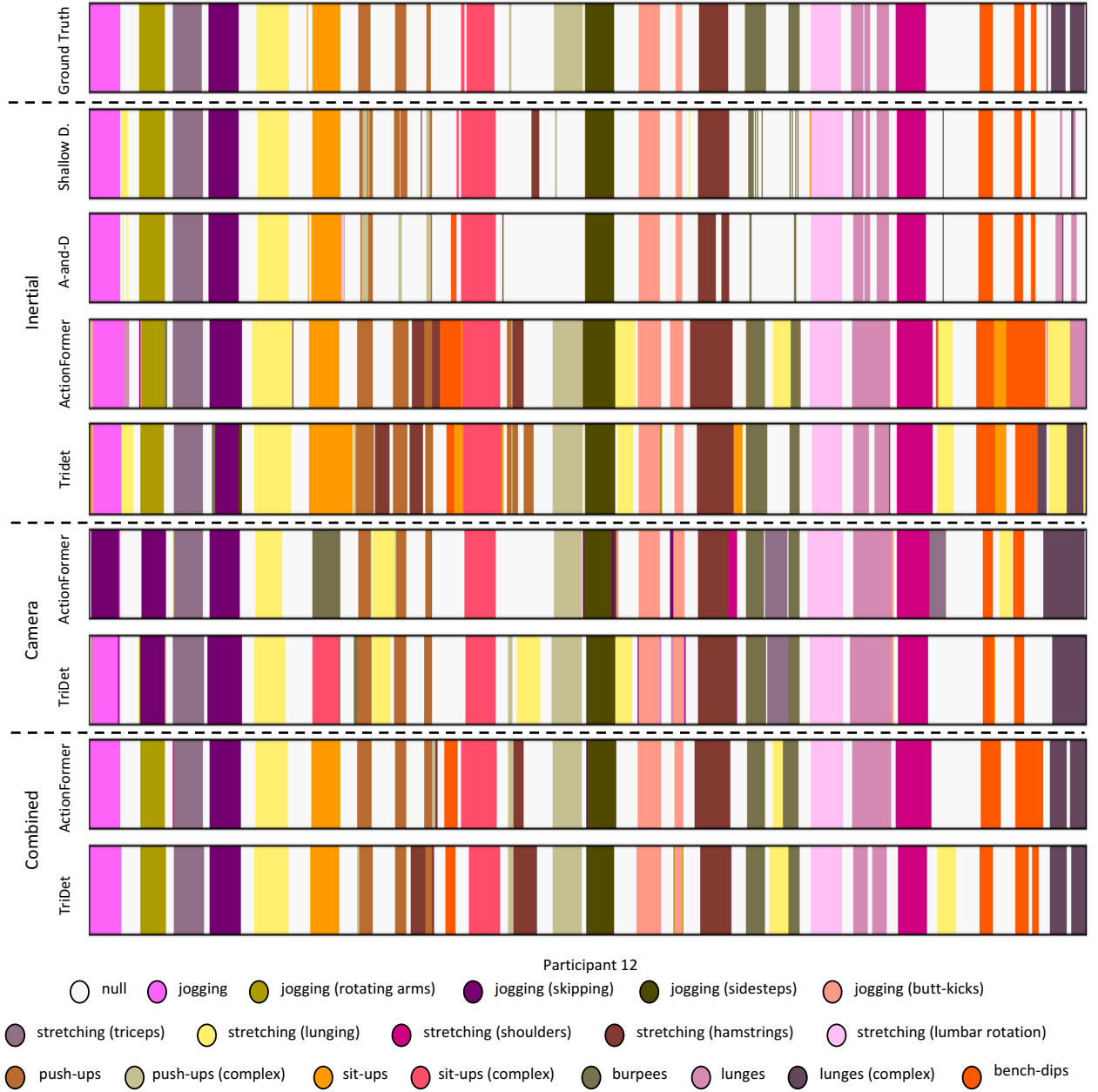


Figure A.8.: Color-coded comparison of the ground truth data (top row) with the shallow DeepConvLSTM, improved Attend-and-Discriminate, ActionFormer and TriDet model on varying input modalities.

A.4. Full-Text Recording Plan

WEAR Dataset Recording Plan

Setup:

- 4 Bangle.js Smartwatches on each limb
 - Sampling rate at 50 Hz with $\pm 8g$
- 2 GoPros
 - Static (GoPro Hero 5):
 - Mounted on tripod; observing whole scene
 - For labeling purposes (not part of final dataset)
 - 1080p 30 FPS Superview FOV (no video stabilization)
 - Head-mounted (GoPro Hero 8):
 - Hands & feet centrally visible in frames; roughly 45° downward
 - 1080p 60 FPS Superview FOV (auto low light + video stabilization)

Equipment List:

- 4 Bangle.js Smartwatches
- 2 GoPros + Head-Mount + Tripod
- Bench/ Box/ Chair
- Optional:
 - Stopwatch
 - Yoga Mat

AFTER RECORDING:
NOTE DOWN THE
BANGLE BUTTON
ORIENTATION & ID!

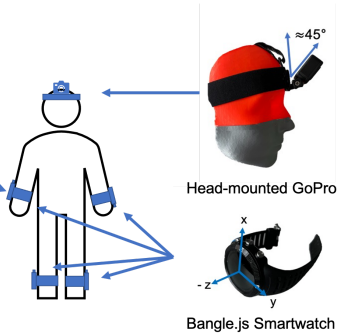


Figure A.9.: First page of the recording plan of the WEAR dataset.

1st Session (ca. 30 min)

→ Each activity 3 sets à repetitions at will (ca. 30 sec); short break after each set (ca. 30 sec)

Activity	Sub-Activity	Description	Normal Variant	Easy Variant
Running	Sidestep	Start with both feet together; jump on one foot to the side and repeat same motion for other foot to create a jumping motion; repeat		
	Butt-Kicks	Fold hands on butt; jog while trying to lift alternating each heel as close to butt as possible ("kicking" it)		
Stretching	Shoulder	Start by standing straight; stretch left arm to the right while keeping it parallel to the ground; use lower right arm to press against left arm upper left arm (close to the elbow) trying to move it closer to the body; hold stretch; repeat by switching arms		
	Hamstrings	Start by sitting down; have left leg stretched out straight in a 45° angle to the left side and keep right leg as sitting cross-legged with right foot touching the left knee's side; try to reach for left foot; hold stretch; repeat by switching legs		
	Lumbar Rotation	Start by laying on back; reach out with both arms to the side; raise legs; move legs to the left side as close as possible to the ground while keeping them straight; do not move torso; hold stretch; repeat by moving legs to opposite side		

Figure A.10.: Second page of the recording plan of the WEAR dataset.

Burpees	Normal	Start by standing straight; put your hands on the ground and jump back with your feet into a push-up position; do a push-up; jump forward with your feet into the starting position and jump up with raised arms; repeat		No Push Up, but lay flat on ground; get up by standing up instead of jumping
Walking Lunges	Normal	Stand straight; Keep your hands crossed in front of your torso; move your right foot forward, bending your left knee to the ground and right knee into a 90° angle; step your left foot forward going back into the position your started in; repeat to create walking motion		
	Complex	Stand straight; keep your arms raised in front of you pointing forward parallel to the floor; do a lunge, but when at the bottom turn your torso to the side of the foot being forward; repeat to create walking motion		
Bench Dips	Normal	Stand with your back facing a chair; go into dip position with your hands on the chair and your legs straight in front of you; move your body downwards by bending your arms into a 90° angle; keep legs and back as straight as possible; move up again; repeat		

Figure A.11.: Third page of the recording plan of the WEAR dataset.

2nd Session (ca. 30 min)

→ Each activity 3 sets à repetitions at will (ca. 30 sec); short break after each set (ca. 30 sec)

Activity	Sub-Activity	Description	Normal Variant	Easy Variant
Running	Jogging	Normal jogging		
	Jogging with rotating arms	Jogging while rotating arms backwards; first only left arm; then only right arm; then both arms simultaneously		
	Skipping	Alternate between jumping from left leg while lifting the right knee high; land on the left leg; repeat by switching legs to create running motion		
	Triceps	Start by standing straight; raise left arm behind head and try to touch the right shoulder; use right arm to grab left elbow trying to move left hand further down the shoulder; hold stretch; repeat by switching arms		
	Lunging	Start by standing in a split stance with right foot forward and left foot straight back; bend right knee about 90°; place hands on your forward knee; hold stretch; repeat by switching legs		

Figure A.12.: Fourth page of the recording plan of the WEAR dataset.

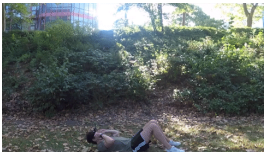
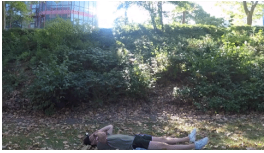
Push-ups	Normal	Normal Push-Up		On knees
	Complex	Move into a push-up position; do a push-up by lowering your body to the ground; after moving it back up, reach out with the right arm to the sky, opening your torso so that it faces to the right; move back into push-up; repeat for left arm; repeat sequence		On knees
Sit-ups	Normal	Lay on your back; have your hands touch the sides of your head; move your legs into a 90° angle with your feet on the ground; move your torso towards your knees while keeping the legs in place; repeat		Straight legs
	Complex	Lay on your back with your hands touching the sides of your head; move your legs into a 90° angle (feet on the ground) while also (!) moving your torso towards your knees; when reaching highest point, touch first your right heel with your right hand; and left heel with left hand; repeat		First move up with upper body; then with lower body so that legs are straight during situp

Figure A.13.: Fifth page of the recording plan of the WEAR dataset.

B. Supplementary Material: Temporal Action Localization for Inertial-based Human Activity Recognition

B.1. Methodological Transparency Appendix

The source code that was used to conduct all experiments, postprocessing and evaluation steps mentioned in this paper is available via Github (https://github.com/mariusbock/tal_for_har). A snapshot of the code is provided as part of the supplementary material download. The repository is written in such a way that other architectures (both inertial- and vision-based) can be added in the future. The repository provides Readme files which give details on the overall structure of the repository, how to add additional datasets and how to set up an Anaconda environment with the needed packages to run experiments. Experiments are defined via '.json'-format configuration files which allow for easy sharing of used hyperparameter settings. All experiments were conducted on a cluster-based system, with each run being assigned one Tesla V100 GPU along with 10 GB of RAM and two AMD EPYC 7452 CPUs. Further, the repository provides links to download each of the six mentioned dataset as used during experiments. Each dataset download is structured into the (1) '.json'-formatted annotations, (2) raw inertial and (3) precomputed vectorized feature embeddings as mentioned in the main paper.

B.2. Supplementary Results of Offline Activity Recognition Experiments

B.2.1. Two-stage Training

Table B.1 provides extended offline activity recognition results of the two-stage training using both LSTM- and attention-based features. Figure B.1 and B.2 present additional confusion matrices of the ActionFormer being applied both with single-stage and two-stage training.

B.2.2. Ablation Study on Postprocessing

The following details ablation experiments conducted to demonstrate the effectiveness and validity of the applied postprocessing described in the experiments section of the main paper.

With the TAL architectures used in this paper being set to predict 2000 segments and are not trained to predict the NULL-class, score thresholding the predicted segments by their

Table B.1.: Average single-stage vs. two-stage LOSO per-sample classification metrics (Precision (P), Recall (R), F1-Score (F1)), misalignment ratios and average mAP applied at different tIoU thresholds (0.3:0.1:07) obtained on six inertial HAR benchmark datasets [142, 140, 149, 19, 77, 164] of the evaluated TAL architectures [204, 167, 151]. Best results per dataset are in **bold**.

	Model	Two-Stage	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	mAP (↑)
Opportunity	ActionFormer		54.63	58.78	51.93	0.19	13.34	0.41	33.82	0.01	0.42	51.24
	TemporalMaxer		44.44	55.63	44.74	0.21	14.13	0.40	43.78	0.01	0.55	46.31
	TriDet		48.72	57.69	48.79	0.23	13.20	0.39	40.07	0.01	0.58	49.70
	ActionFormer	L	60.58	38.29	40.85	0.26	10.38	0.78	26.43	0.00	0.46	29.81
	TemporalMaxer	L	59.52	40.10	41.49	0.28	12.48	0.73	27.26	0.01	0.65	29.44
	TriDet	L	60.80	39.25	41.47	0.27	10.26	0.76	27.78	0.00	0.51	30.77
	ActionFormer	A	63.97	53.24	53.41	0.24	12.55	0.52	24.10	0.00	0.77	46.71
	TemporalMaxer	A	60.63	54.43	52.98	0.27	15.70	0.46	24.39	0.01	0.91	43.50
	TriDet	A	64.29	54.31	54.74	0.25	11.96	0.49	24.23	0.01	0.87	47.17
SBHAR	ActionFormer		87.02	84.37	84.43	0.36	5.41	0.16	5.09	0.00	0.20	95.46
	TemporalMaxer		86.52	83.37	83.66	0.41	6.02	0.19	4.41	0.00	0.24	94.39
	TriDet		88.95	86.15	86.45	0.38	5.41	0.12	3.95	0.01	0.29	94.75
	ActionFormer	L	86.52	76.49	78.96	0.38	3.16	0.35	6.72	0.00	0.04	84.82
	TemporalMaxer	L	85.66	75.53	78.04	0.42	3.22	0.37	6.88	0.00	0.14	84.38
	TriDet	L	85.89	76.87	79.15	0.36	2.94	0.34	7.68	0.01	0.17	83.65
	ActionFormer	A	90.00	85.69	86.60	0.33	4.60	0.19	4.64	0.00	0.02	93.31
	TemporalMaxer	A	89.21	83.87	85.02	0.37	4.95	0.20	4.93	0.00	0.13	93.23
	TriDet	A	90.04	86.83	87.23	0.30	4.88	0.20	4.31	0.01	0.10	93.50
Wetlab	ActionFormer		40.71	49.25	40.71	0.56	9.41	0.79	52.44	0.07	0.79	33.53
	TemporalMaxer		50.43	36.65	37.09	0.59	9.37	0.83	53.59	0.10	0.61	35.72
	TriDet		44.13	49.15	42.85	0.58	8.85	0.75	48.63	0.10	0.84	34.05
	ActionFormer	L	47.87	43.36	43.06	0.69	4.66	0.93	36.92	0.05	0.59	26.61
	TemporalMaxer	L	46.68	44.54	42.87	0.70	4.96	0.91	40.09	0.08	0.58	25.66
	TriDet	L	47.37	43.10	42.66	0.65	4.08	0.95	35.95	0.07	0.79	27.06
	ActionFormer	A	47.01	56.12	47.30	0.55	9.69	0.64	40.46	0.06	1.09	43.07
	TemporalMaxer	A	44.14	56.96	45.36	0.51	9.73	0.67	44.92	0.08	1.03	40.55
	TriDet	A	48.00	54.51	47.57	0.53	7.94	0.71	39.78	0.07	1.00	41.20
WEAR	ActionFormer		71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
	TemporalMaxer		69.54	72.80	69.52	0.23	6.87	0.83	5.50	0.01	1.50	69.18
	TriDet		73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
	ActionFormer	L	80.32	76.63	75.99	0.27	2.46	0.49	8.15	0.12	0.66	63.38
	TemporalMaxer	L	79.61	77.31	75.80	0.28	3.50	0.49	7.94	0.10	0.90	59.23
	TriDet	L	80.18	76.63	75.65	0.29	3.22	0.47	8.24	0.13	0.55	61.41
	ActionFormer	A	85.01	86.34	84.56	0.20	4.51	0.30	2.94	0.01	1.42	85.17
	TemporalMaxer	A	82.04	84.31	82.13	0.20	4.81	0.39	3.25	0.01	1.40	82.97
	TriDet	A	86.29	85.77	84.78	0.24	3.35	0.29	3.39	0.02	1.33	85.94
Hang-Time	ActionFormer		49.19	57.57	51.23	0.62	11.63	0.51	47.65	0.48	0.64	29.26
	TemporalMaxer		45.01	54.56	47.17	0.71	10.97	0.45	52.71	1.13	0.65	27.86
	TriDet		49.59	55.14	50.67	0.73	9.85	0.52	48.55	0.69	0.62	29.24
	ActionFormer	L	56.10	51.63	51.76	0.57	8.90	0.57	42.21	0.31	0.18	23.80
	TemporalMaxer	L	54.92	51.99	51.51	0.57	8.87	0.53	43.87	0.41	0.23	23.72
	TriDet	L	56.72	50.55	51.43	0.59	8.39	0.60	41.82	0.35	0.20	22.70
	ActionFormer	A	53.64	62.47	55.81	0.55	11.13	0.39	43.25	0.28	0.50	37.52
	TemporalMaxer	A	51.39	61.24	53.79	0.56	11.18	0.36	45.52	0.72	0.47	36.77
	TriDet	A	54.83	62.27	56.66	0.59	9.50	0.39	43.40	0.26	0.50	36.84
RWHAR	ActionFormer		63.76	67.64	61.24	2.48	11.12	2.06	11.23	0.09	0.00	65.40
	TemporalMaxer		63.20	67.60	60.59	2.59	11.95	1.81	13.96	0.36	0.05	50.60
	TriDet		69.27	73.04	67.86	1.48	6.88	2.03	10.24	0.23	0.00	69.98
	ActionFormer	L	71.95	74.76	69.67	1.97	10.63	1.56	6.87	0.16	0.00	71.37
	TemporalMaxer	L	75.35	77.72	72.61	1.55	7.76	1.33	8.00	0.37	0.00	39.79
	TriDet	L	77.30	80.00	75.56	1.23	6.07	1.37	7.47	0.29	0.00	73.53
	ActionFormer	A	80.32	80.94	78.09	1.81	9.16	0.87	5.24	0.16	0.00	75.07
	TemporalMaxer	A	76.87	78.10	73.66	2.11	9.41	1.07	7.54	0.20	0.06	60.77
	TriDet	A	85.82	86.86	84.08	0.96	4.64	0.78	5.09	0.21	0.00	77.86

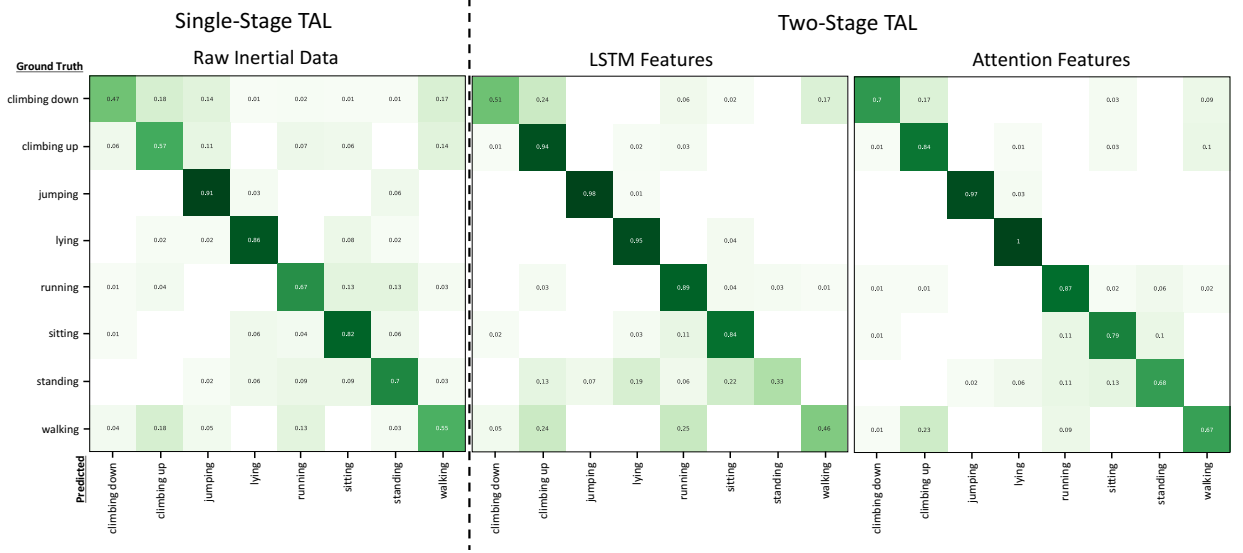


Figure B.1.: Single-Stage vs. Two-Stage confusion matrices of the ActionFormer [204] applied on the RWHAR dataset [164]. Note that confusions which are 0 are omitted.

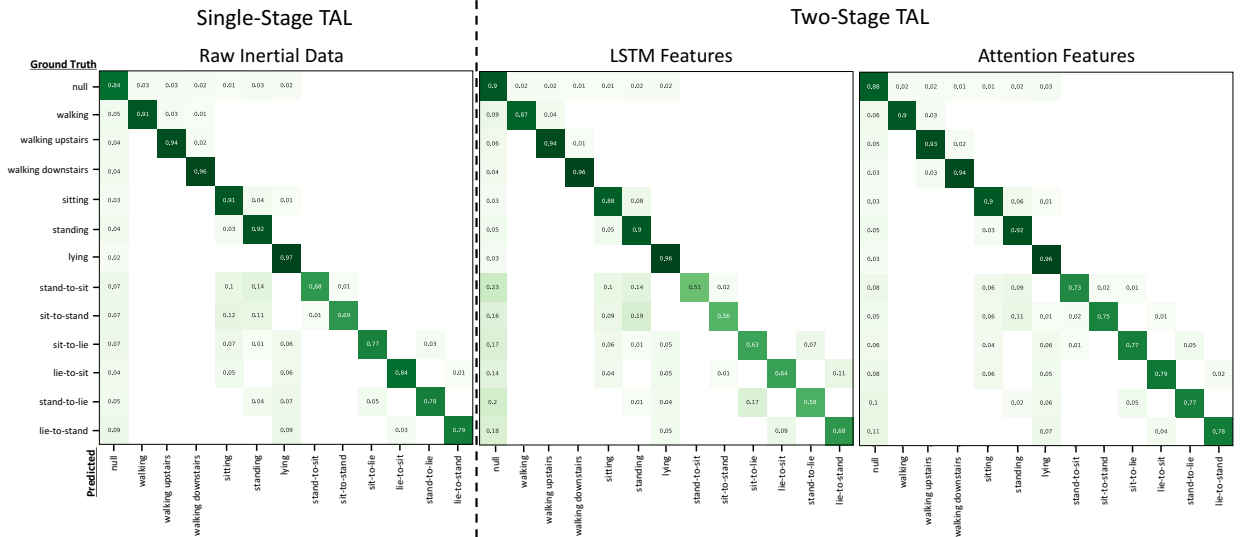


Figure B.2.: Single-Stage vs. Two-Stage confusion matrices of the ActionFormer [204] applied on the SBHAR dataset [140]. Note that confusions which are 0 are omitted.

associated confidence is essential in order to preserve breaks. As evident by results presented in Table B.2 and Figure B.3, without score thresholding (almost) all timestamps get classified into one of the activity classes. Applying a score threshold thus results in significant increases in NULL-class accuracy across all datasets. Though mAP scores decrease, overall F1-scores increase by a significant margin across all datasets. The optimal score threshold for each dataset was chosen by trying to balance the increase in F1-score compared to the decrease in mAP score.

The inertial architectures in this paper are optimized to predict per-window class labels. This causes unprocessed prediction streams of all three models to show frequently occurring, short activity switches as the context of prior windows is disregarded during the prediction of the current timestamp. This causes segments of activities to get fragmented frequently, ultimately leading the mAP score of the inertial models to be significantly lower than that of TAL architectures. Therefore, all prediction streams of the inertial architectures were smoothed using a majority filter. The exact size of each majority filter was chosen dataset-dependent, trying out different filter sizes and choosing the one which balanced changes in F1-score and mAP to the best extend. Table B.3 and Figure B.4 illustrate the effect of the majority filter on the inertial architecture’s prediction streams.

B.2.3. Additional Confusion Matrices

The following provides additional confusion matrices which were not part of the results chapter. Specifically, we provide confusion matrices of the TAL and inertial architectures being applied on the Opportunity [142] (see Figures B.5 and B.5), the Wetlab [149] (see Figures B.7 and B.8), the WEAR [19] (see Figures B.9 and B.10) and the Hang-Time [77] (see Figures B.11 and B.12) dataset. All experiments were conducted as specified in the main paper.

Table B.2.: Unprocessed versus postprocessed average LOSO results obtained on the six inertial HAR benchmark datasets [142, 140, 149, 19, 77, 164] using the three temporal action localization models [204, 167, 151]. The table provides along the applied score threshold s the per-sample classification metrics, i.e. Precision (P), Recall (R), F1-Score (F1), misalignment ratios [182] and average mAP applied at different tIoU thresholds (0.3:0.1:07). All experiments were conducted the same style as reported in the main paper. Best results per dataset are in **bold**.

	s	Model	P ($\uparrow$)	R ($\uparrow$)	F1 ($\uparrow$)	UR ($\downarrow$)	OR ($\downarrow$)	DR ($\downarrow$)	IR ($\downarrow$)	FR ($\downarrow$)	MR ($\downarrow$)	mAP ($\uparrow$)
Opportunity	0.0	ActionFormer	48.08	61.44	49.37	0.20	13.83	0.35	40.38	0.01	0.43	53.74
		TemporalMaxer	41.34	56.50	43.11	0.21	13.85	0.37	47.41	0.01	0.56	47.32
		TriDet	41.41	59.30	44.61	0.25	13.11	0.31	48.08	0.01	0.54	51.97
	0.2	ActionFormer	54.63	58.78	51.93	0.19	13.34	0.41	33.82	0.01	0.42	51.24
		TemporalMaxer	44.44	55.63	44.74	0.21	14.13	0.40	43.78	0.01	0.55	46.31
		TriDet	48.72	57.69	48.79	0.23	13.20	0.39	40.07	0.01	0.58	49.70
SBHAR	0.0	ActionFormer	60.16	79.85	66.84	0.26	10.05	0.09	24.13	0.02	0.63	97.08
		TemporalMaxer	61.96	81.18	68.66	0.29	10.21	0.09	23.80	0.02	0.61	96.28
		TriDet	61.96	81.18	68.66	0.28	7.93	0.06	24.15	0.04	0.61	97.16
	0.3	ActionFormer	87.02	84.37	84.43	0.36	5.41	0.16	5.09	0.00	0.20	95.46
		TemporalMaxer	86.52	83.37	83.66	0.41	6.02	0.19	4.41	0.00	0.24	94.39
		TriDet	88.95	86.15	86.45	0.38	5.41	0.12	3.95	0.01	0.29	94.75
Wetlab	0.0	ActionFormer	27.54	51.37	23.74	0.59	7.99	0.53	71.98	0.16	0.54	39.56
		TemporalMaxer	27.98	50.02	23.65	0.62	7.47	0.59	72.28	0.21	0.44	38.68
		TriDet	27.92	51.82	24.12	0.60	6.85	0.43	73.41	0.24	0.54	41.72
	0.15	ActionFormer	40.71	49.25	40.71	0.56	9.41	0.79	52.44	0.07	0.79	33.53
		TemporalMaxer	50.43	36.65	37.09	0.59	9.37	0.83	53.59	0.10	0.61	35.72
		TriDet	44.13	49.15	42.85	0.58	8.85	0.75	48.63	0.10	0.84	34.05
WEAR	0.0	ActionFormer	56.87	78.35	60.81	0.18	9.13	0.46	26.22	0.02	2.76	81.63
		TemporalMaxer	56.72	79.76	61.70	0.18	8.51	0.42	28.13	0.03	2.49	78.69
		TriDet	53.97	78.22	60.63	0.24	6.52	0.36	31.61	0.09	2.22	83.08
	0.2	ActionFormer	71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
		TemporalMaxer	69.54	72.80	69.52	0.23	6.87	0.83	5.50	0.01	1.50	69.18
		TriDet	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
Hang-Time	0.0	ActionFormer	41.26	58.23	44.11	0.54	10.75	0.33	58.70	0.42	0.63	31.21
		TemporalMaxer	39.21	56.20	42.07	0.55	9.73	0.36	61.94	0.79	0.60	29.23
		TriDet	39.98	57.17	43.05	0.55	8.40	0.31	62.55	0.54	0.58	31.75
	0.15	ActionFormer	49.19	57.57	51.23	0.62	11.63	0.51	47.65	0.48	0.64	29.26
		TemporalMaxer	45.01	54.56	47.17	0.71	10.97	0.45	52.71	1.13	0.65	27.86
		TriDet	49.59	55.14	50.67	0.73	9.85	0.52	48.55	0.69	0.62	29.24

Table B.3.: Unprocessed versus postprocessed average LOSO results obtained on the six inertial HAR benchmark datasets [142, 140, 149, 19, 77, 164] using the four inertial architectures [126, 17, 1, 210]. The table provides along the applied majority filter m the per-sample classification metrics, i.e. Precision (P), Recall (R), F1-Score (F1), misalignment ratios [182] and average mAP applied at different tIoU thresholds (0.3:0.1:0.7). Best results per dataset are in **bold**.

	m	Model	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	mAP (↑)
Opportunity	0 sec.	DeepConvLSTM	45.29	34.91	34.75	0.50	9.61	0.49	47.27	0.09	0.12	6.16
		Shallow D.	39.62	27.93	27.33	0.41	8.88	0.71	52.50	0.08	0.09	4.75
		A-and-D	30.61	45.61	32.87	0.46	7.89	0.27	65.02	0.10	0.17	5.63
		TinyHAR	42.12	51.65	43.26	0.46	11.37	0.23	49.12	0.09	0.29	9.35
	2.5 sec.	DeepConvLSTM	50.22	33.88	34.41	0.30	17.29	0.78	31.26	0.01	0.23	13.97
		Shallow D.	42.08	27.18	26.46	0.24	14.28	0.97	35.06	0.01	0.15	10.61
		A-and-D	35.25	48.55	36.35	0.32	15.28	0.45	52.55	0.02	0.35	13.75
		TinyHAR	48.09	54.27	47.09	0.34	19.99	0.39	34.01	0.02	0.46	19.78
	0 sec.	DeepConvLSTM	54.04	59.98	54.67	0.61	3.31	0.29	44.48	0.51	0.00	18.58
		Shallow D.	72.29	75.80	70.89	0.65	8.58	0.34	19.20	0.13	0.02	52.03
		A-and-D	55.08	65.19	56.12	0.51	3.83	0.25	43.47	0.40	0.01	21.99
		TinyHAR	50.91	59.43	50.48	0.59	3.20	0.25	48.60	0.47	0.00	16.32
	5 sec.	DeepConvLSTM	67.54	63.72	62.31	0.41	7.19	0.59	21.44	0.07	0.10	49.60
		Shallow D.	72.98	75.41	71.13	0.60	10.19	0.46	14.23	0.02	0.09	65.15
		A-and-D	68.64	71.07	66.49	0.31	9.47	0.45	21.71	0.05	0.11	55.79
		TinyHAR	58.91	63.70	56.29	0.31	8.16	0.54	31.67	0.05	0.13	45.38
SBHAR	0 sec.	DeepConvLSTM	25.62	38.69	26.31	0.48	0.84	0.10	81.78	1.47	0.00	0.45
		Shallow D.	38.4	37.29	35.01	0.79	6.77	1.04	57.65	0.28	0.14	6.08
		A-and-D	27.38	44.65	26.60	0.51	1.16	0.06	79.95	1.37	0.00	0.65
		TinyHAR	26.76	42.84	24.44	0.68	1.28	0.15	80.51	1.26	0.00	1.22
	20 sec.	DeepConvLSTM	38.65	47.34	37.87	0.51	7.13	0.69	48.53	0.21	0.64	11.88
		Shallow D.	39.01	35.42	34.42	0.51	9.52	1.57	34.51	0.06	0.43	15.40
		A-and-D	37.75	55.71	37.49	0.56	9.69	0.63	57.60	0.16	0.57	12.27
		TinyHAR	34.31	50.84	31.48	0.61	8.41	1.00	61.26	0.11	0.59	10.05
	0 sec.	DeepConvLSTM	74.86	70.40	70.04	0.21	1.06	0.18	22.16	0.73	0.02	6.57
		Shallow D.	82.94	77.99	77.09	0.31	2.55	0.28	9.60	0.26	0.07	36.42
		A-and-D	73.37	76.71	72.87	0.16	1.51	0.08	25.03	0.61	0.05	9.08
		TinyHAR	75.33	81.72	76.78	0.13	1.82	0.06	22.95	0.46	0.05	12.41
	15 sec.	DeepConvLSTM	80.68	76.25	75.78	0.28	2.35	0.52	6.68	0.11	0.32	61.03
		Shallow D.	80.78	78.91	77.71	0.27	3.23	0.50	5.21	0.04	0.43	67.89
		A-and-D	82.34	83.29	80.61	0.20	4.03	0.33	7.18	0.09	0.52	64.78
		TinyHAR	83.63	87.79	83.96	0.15	5.09	0.20	7.87	0.09	0.54	68.97
Wetlab	0 sec.	DeepConvLSTM	34.35	39.84	35.82	0.61	4.75	0.21	53.94	0.91	0.03	2.91
		Shallow D.	38.12	44.24	39.20	0.57	9.29	0.57	44.73	0.39	0.63	5.95
		A-and-D	32.48	45.94	35.14	0.63	4.39	0.17	56.78	0.98	0.04	3.41
		TinyHAR	29.99	42.25	31.80	0.65	3.69	0.23	60.35	1.20	0.01	2.60
	5 sec.	DeepConvLSTM	44.13	33.95	35.25	0.28	10.60	0.88	20.16	0.27	1.46	5.44
		Shallow D.	37.97	38.19	36.85	0.35	14.00	1.07	36.38	0.21	2.33	5.00
		A-and-D	40.54	43.32	40.39	0.35	14.14	0.72	34.81	0.30	1.44	6.73
		TinyHAR	37.09	41.13	36.89	0.41	11.59	0.75	42.90	0.43	1.26	4.73
	0 sec.	DeepConvLSTM	74.22	75.27	71.35	0.57	0.24	0.19	24.99	3.01	0.00	0.00
		Shallow D.	89.89	87.40	85.53	0.43	0.44	0.04	9.39	1.43	0.00	0.00
		A-and-D	75.17	76.30	72.25	0.59	0.32	0.30	23.67	2.70	0.00	0.00
		TinyHAR	80.90	80.94	77.86	0.46	0.26	0.04	18.84	2.46	0.00	0.00
	40 sec.	DeepConvLSTM	79.05	81.93	77.56	0.65	2.05	1.09	13.63	1.07	0.00	0.11
		Shallow D.	88.59	89.01	86.85	0.31	1.14	0.38	6.93	0.98	0.00	0.00
		A-and-D	79.46	82.68	78.09	0.66	2.31	1.17	11.28	0.84	0.00	0.04
		TinyHAR	83.59	86.25	82.62	0.53	1.14	0.76	11.65	0.89	0.00	0.02
RW-HAR	0 sec.	DeepConvLSTM	79.05	81.93	77.56	0.65	2.05	1.09	13.63	1.07	0.00	0.11
		Shallow D.	88.59	89.01	86.85	0.31	1.14	0.38	6.93	0.98	0.00	0.00
		A-and-D	79.46	82.68	78.09	0.66	2.31	1.17	11.28	0.84	0.00	0.04
		TinyHAR	83.59	86.25	82.62	0.53	1.14	0.76	11.65	0.89	0.00	0.02

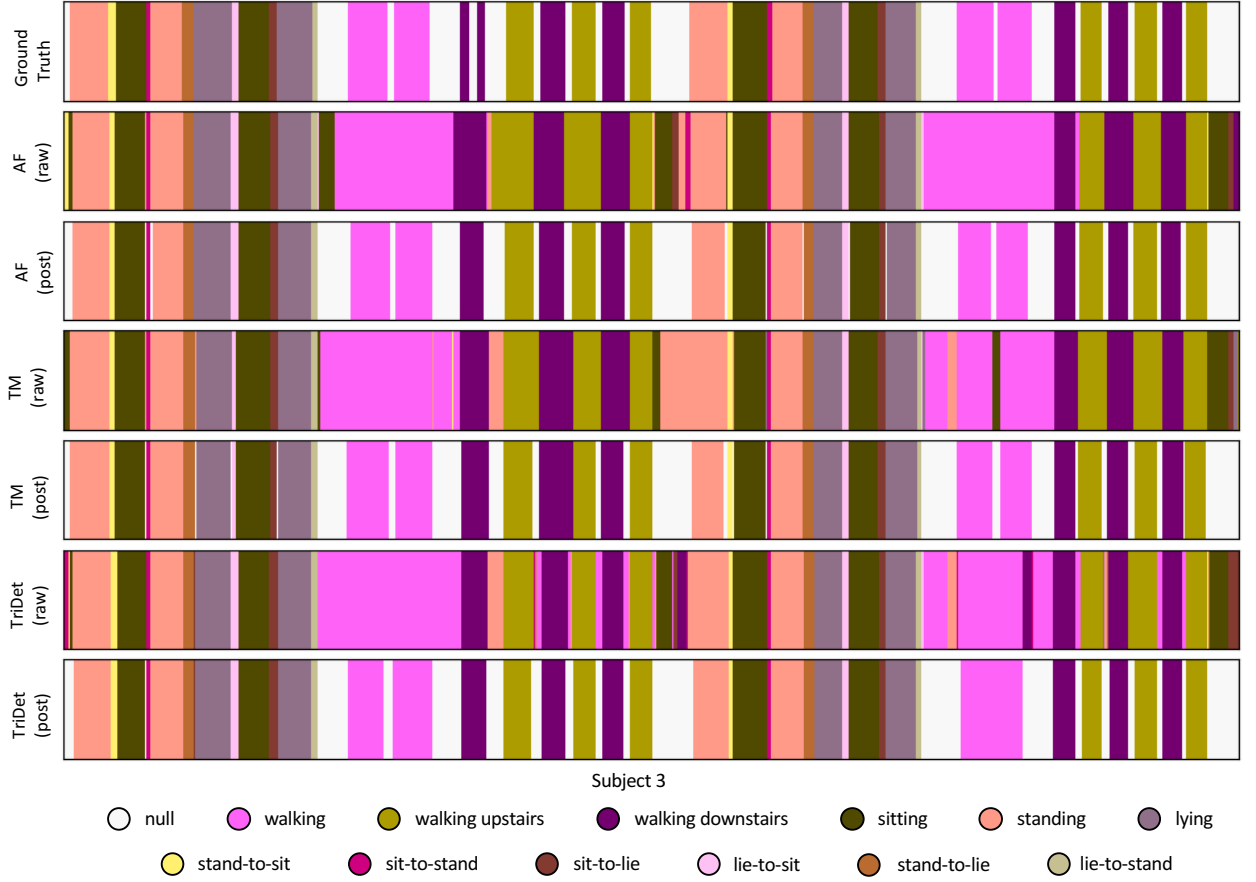


Figure B.3.: Colored visualization of unprocessed versus postprocessed prediction streams obtained on the inertial SBHAR dataset [140] using the three TAL architectures [204, 167, 151]. Postprocessing was performed using majority filter of 15 seconds. All experiments were conducted the same style as reported in the main paper. One can see that score thresholding predictions causes not every timestamp to be predicted and NULL-class accuracy to rise. Visualized results are that of Subject 3.

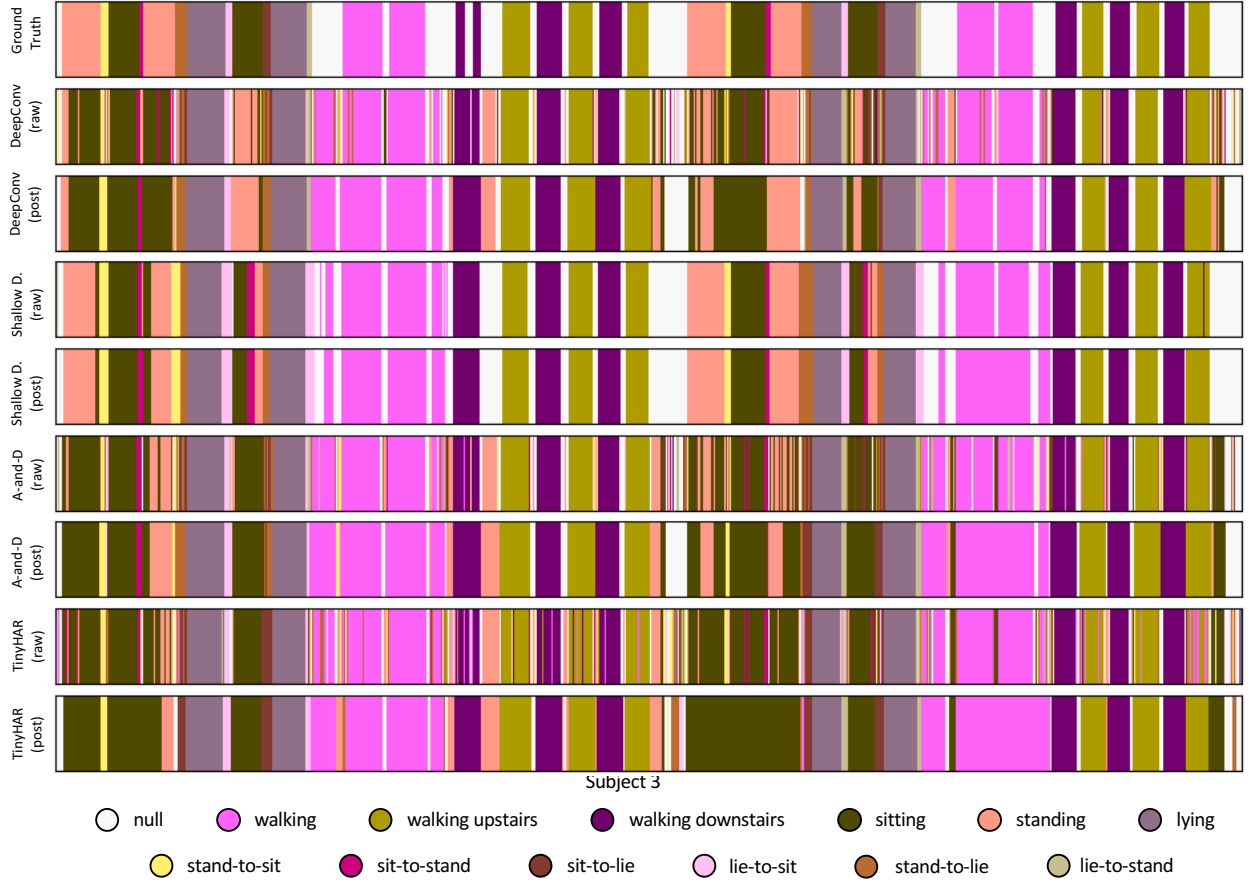


Figure B.4.: Colored visualization of unprocessed versus postprocessed prediction streams obtained on the inertial SBHAR dataset [140] using the four inertial architectures [126, 17, 1, 210]. Postprocessing was performed using majority filter of 15 seconds. All experiments were conducted the same style as reported in the main paper. One can see that majority filtering the predictions causes short activity switches to disappear with more coherent segments being created. Visualized results are that of Subject 3.

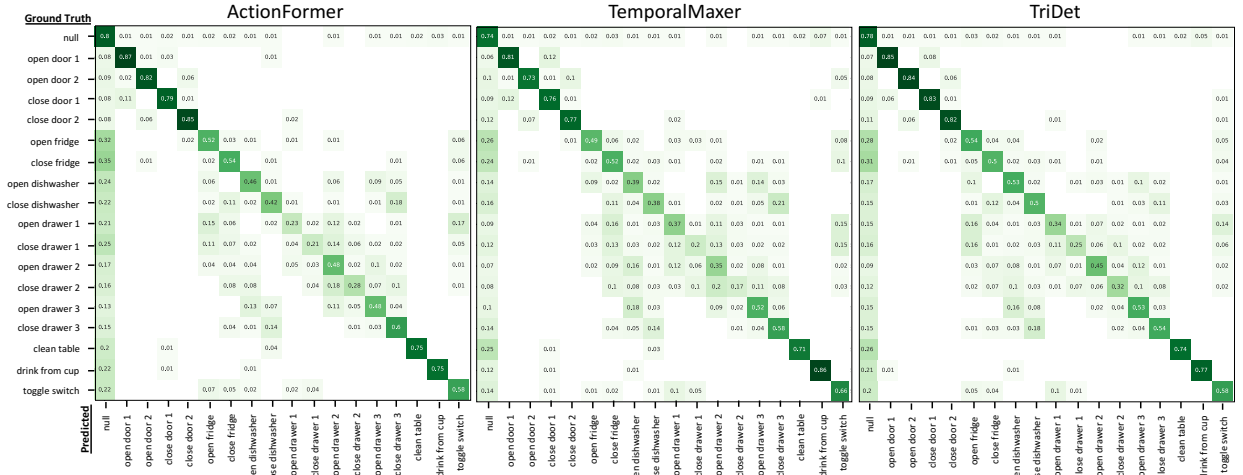


Figure B.5.: Confusion matrices of the ActionFormer [204], TemporalMaxer [167] and TriDet model [151] applied on the Opportunity dataset [142] with a one second sliding window and 50% overlap. Results are postprocessed using a score threshold of 0.2. Note that confusions which are 0 are omitted.

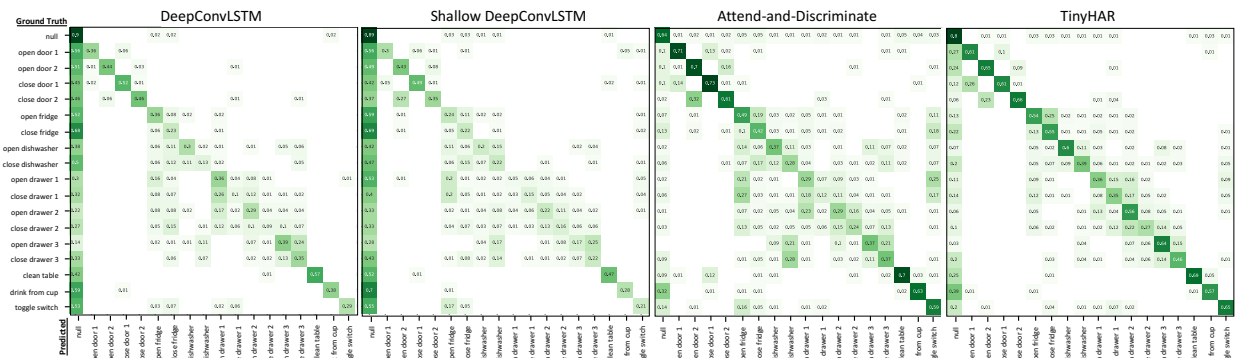


Figure B.6.: Confusion matrices of the DeepConvLSTM [126], shallow DeepConvLSTM [17], Attend-and-Discriminate [1] and TinyHAR model [210] applied on the Opportunity dataset [142] with a one second sliding window and 50% overlap. Results are postprocessed using a majority filter of 2.5 seconds. Note that confusions which are 0 are omitted.

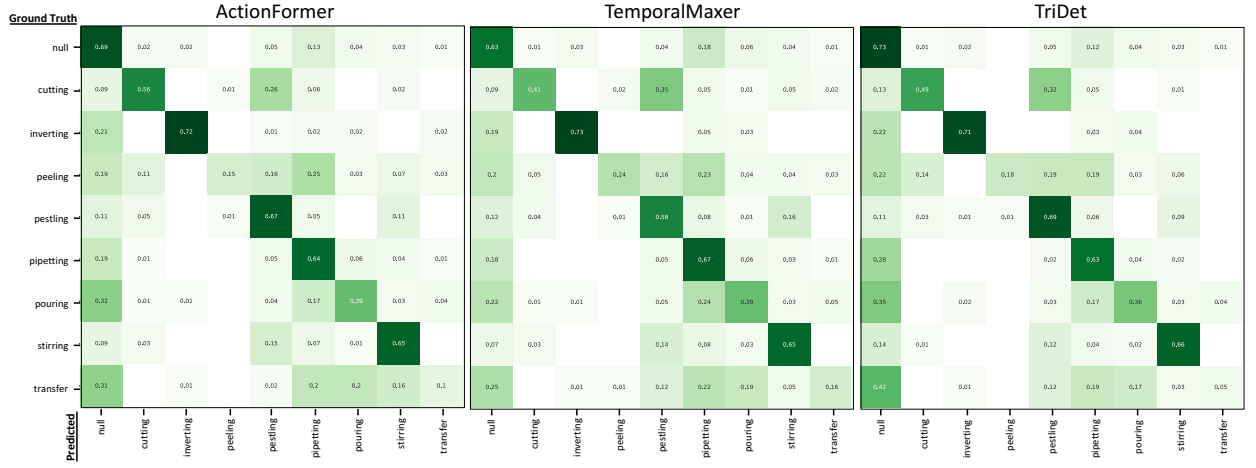


Figure B.7.: Confusion matrices of the ActionFormer [204], TemporalMaxer [167] and TriDet model [151] applied on the Wetlab dataset [149] with a one second sliding window and 50% overlap. Results are post-processed using a score threshold of 0.15. Note that confusions which are 0 are omitted.

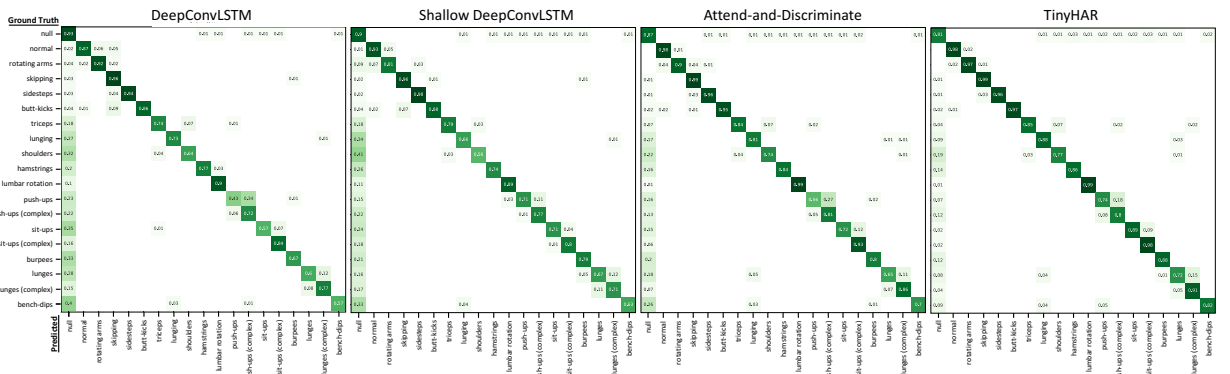


Figure B.8.: Confusion matrices of the DeepConvLSTM [126], shallow DeepConvLSTM [17], Attend-and-Discriminate [1] and TinyHAR model [210] applied on the Wetlab dataset [149] with a one second sliding window and 50% overlap. Results are postprocessed using a majority filter 20 seconds. Note that confusions which are 0 are omitted.

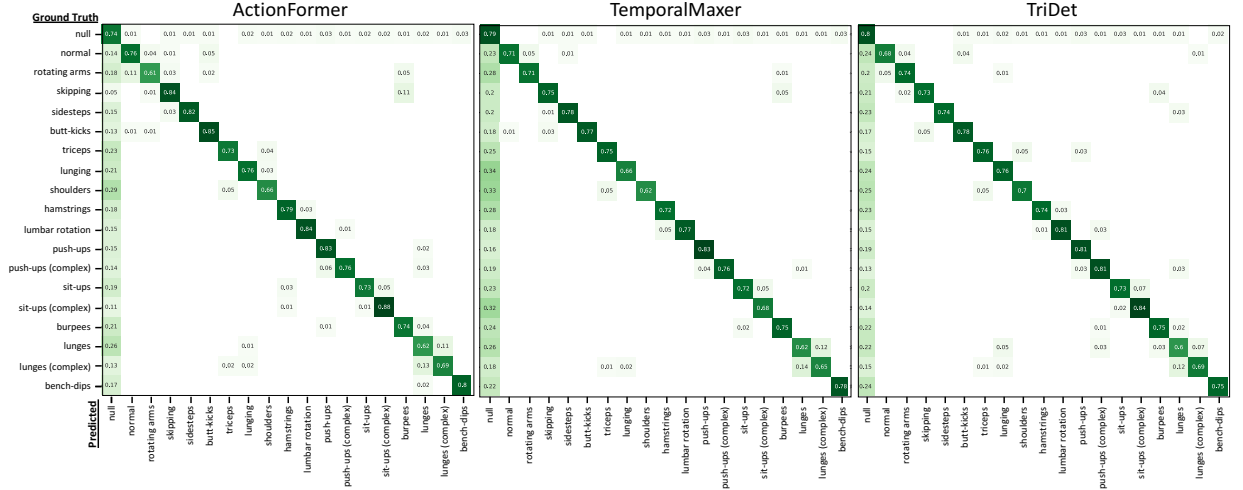


Figure B.9.: Confusion matrices of the ActionFormer [204], TemporalMaxer [167] and TriDet model [151] applied on the WEAR dataset [19] with a one second sliding window and 50% overlap. Compared to inertial architectures such as the DeepConvLSTM [126] transition activities get more reliably predicted. Results are postprocessed using a score threshold of 0.2. Note that confusions which are 0 are omitted.

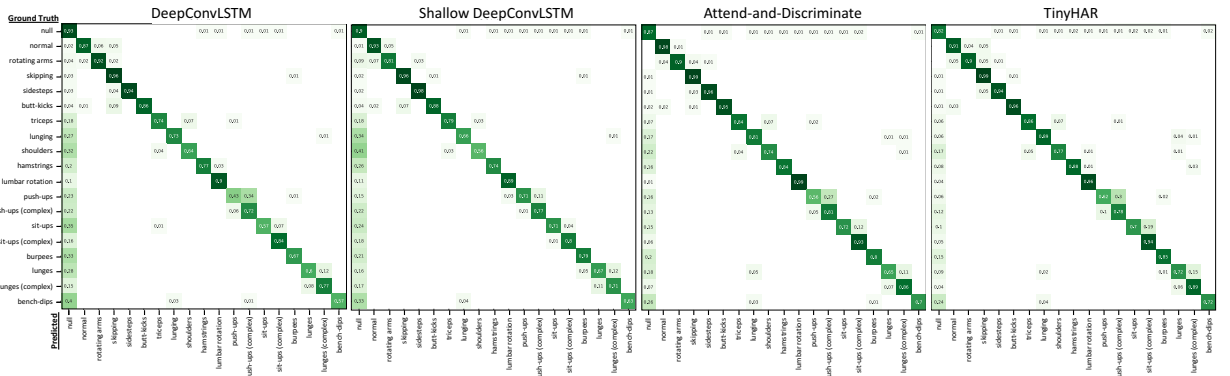


Figure B.10.: Confusion matrices of the DeepConvLSTM [126], shallow DeepConvLSTM [17], Attend-and-Discriminate [1] and TinyHAR model [210] applied on the WEAR dataset [19] with a one second sliding window and 50% overlap. Results are postprocessed using a majority filter 15 seconds. Note that confusions which are 0 are omitted.

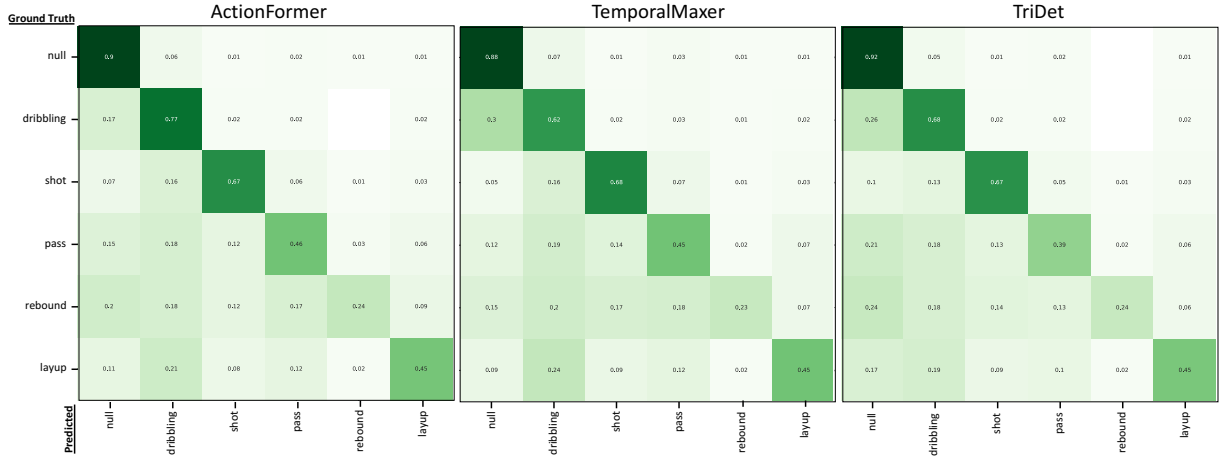


Figure B.11.: Confusion matrices of the ActionFormer [204], TemporalMaxer [167] and TriDet model [151] applied on the Hang-Time dataset [77] with a one second sliding window and 50% overlap. Results are postprocessed using a score threshold of 0.15. Note that confusions which are 0 are omitted.

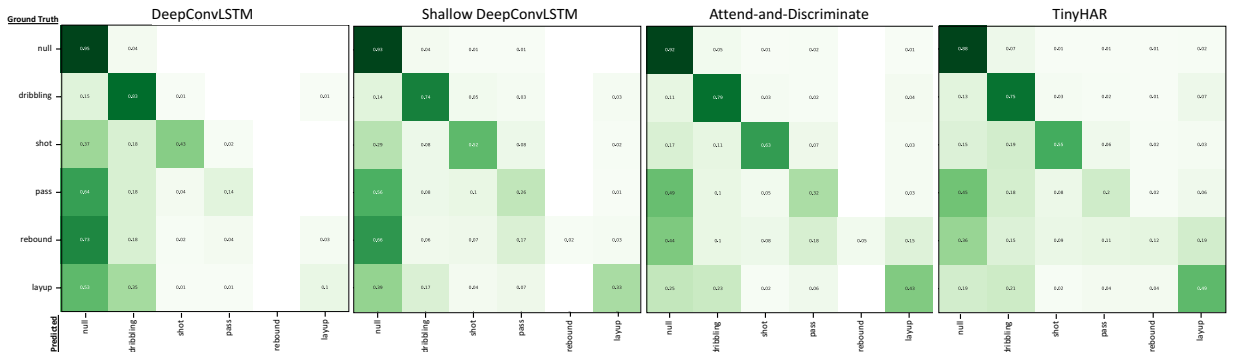


Figure B.12.: Confusion matrices of the DeepConvLSTM [126], shallow DeepConvLSTM [17], Attend-and-Discriminate [1] and TinyHAR model [210] applied on the Hang-Time dataset [77] with a one second sliding window and 50% overlap. Results are postprocessed using a majority filter 15 seconds. Note that confusions which are 0 are omitted.

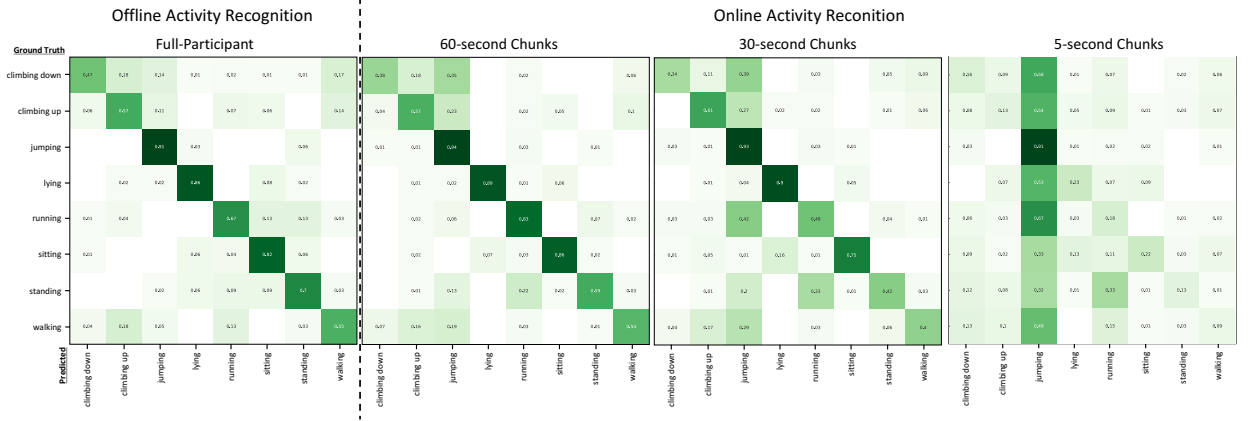


Figure B.13.: Offline vs. online prediction results presented as confusion matrices of the ActionFormer [204] applied on the RWHAR dataset [164]. Illustrated are results applying a chunking of 60, 30 and 5 seconds. Note that confusions which are 0 are omitted.

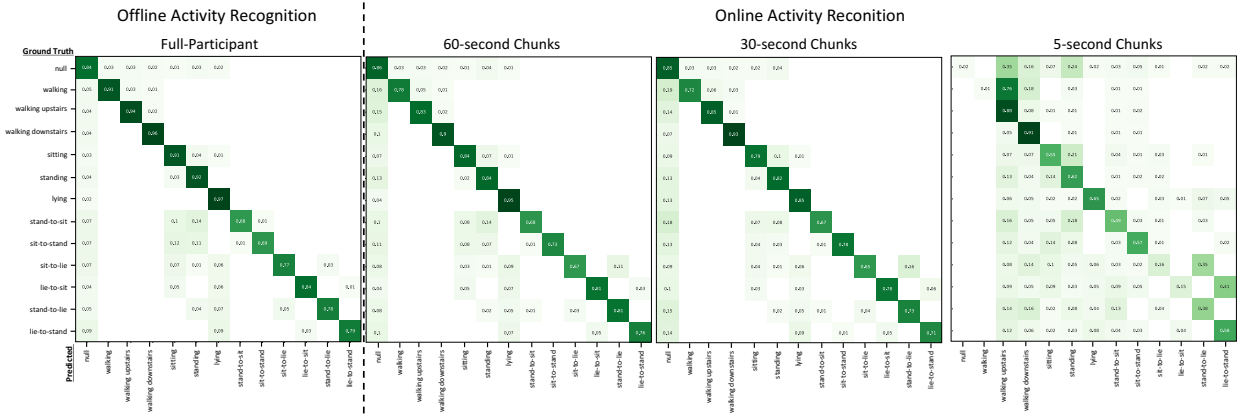


Figure B.14.: Offline vs. online prediction results presented as confusion matrices of the ActionFormer [204] applied on the SBHAR dataset [140]. Illustrated are results applying a chunking of 60, 30 and 5 seconds. Note that confusions which are 0 are omitted.

B.3. Supplementary Results of Online Activity Recognition

Figure B.13 and B.14 present additional confusion matrices of the ActionFormer being applied in an online prediction scenario.

B.4. DETAD Analysis

In 2018 Alwassel et al. [6] released a tool called Diagnosing Error in Temporal Action Detectors (DETAD). The tool, targeted towards analyzing temporal action localization algorithms consists of three analyses: a false positive analysis, a localization metric sensitivity analysis, and a false negative analysis. Similar to Ward et al. [181]’s efforts in the inertial community,

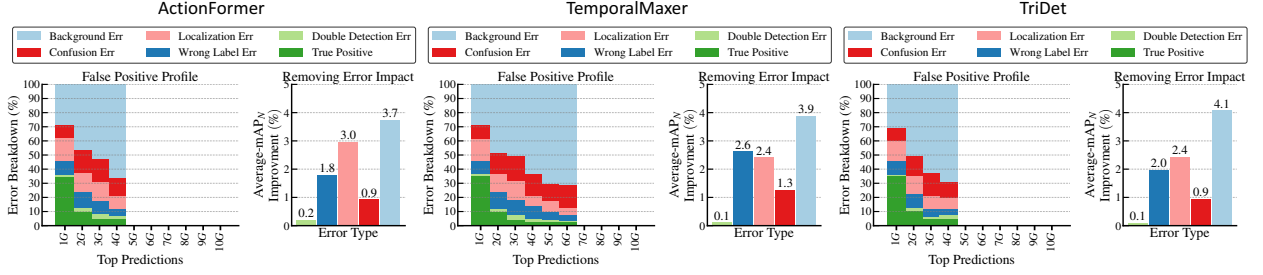


Figure B.15.: DETAD False Positive (FP) analysis of the TAL architectures [204, 167, 151] on the Wetlab dataset [149]. Experiments are those described in the results chapter of the main paper with the random seed being fixed at 1. Each subplot exists of a FP error breakdown (left) and an impact analysis of error (right). The FP error breakdown is divided considering the top-10 predictions. One can see that the background error and localization error are most crucial. Furthermore, the TemporalMaxer’s predictions show a larger overlap amongst each other with there being a higher degree of top-N predictions.

DETAD is supposed to provide an easy-to-use tool which analyzes the performance of temporal action detectors in videos beyond a single scalar metric, i.e. mAP. Given a dataset, DETAD divides segments into bins according to three criterion: coverage, length, and the number of instances. According to coverage, being the relative size of a segment compared to the length of the complete sequence, each segment can be divided into five bins: XS: (0, 0.02], S: (0.02, 0.04], M: (0.04, 0.06], L: (0.06, 0.08], and XL: (0.08, 1.0]. Length, being the absolute length (in seconds) of segments, can be divided into five bins: XS: (0 seconds, 3 seconds], S: (3 seconds, 6 seconds], M: (6 seconds, 12 seconds], L: (12 seconds, 18 seconds], and XL: more than 18 seconds. Number of instances refers to the number of segments of the same class within a video. This can be divided into five bins: XS: 1, Small S: [2, 40], M: [40, 80], and L: ≥ 80 . Our repository integrates the scripts provided by Alwassel et al. [6] such that the DETAD analysis can easily be run on both inertial and temporal action localization models. The following provides sample results obtained on the Wetlab dataset [149]. All other plots created by the three DETAD analyses performed on the other datasets can be found in the repository.

In general, the DETAD analyses reveal that TAL models are less prone to background errors than inertial models. Furthermore, results obtained on the shallow DeepConvLSTM shows similarities to that of the TAL models, further suggesting that the inertial model is capable of learning temporal dependencies across windows. In case of the Wetlab dataset the three types of analysis (see Figure B.15, B.16 and B.17) show that the TAL models especially struggle with predicting small segments correctly. In general, differentiation with the null class as well localizing the activities is one of the largest error sources for both models.

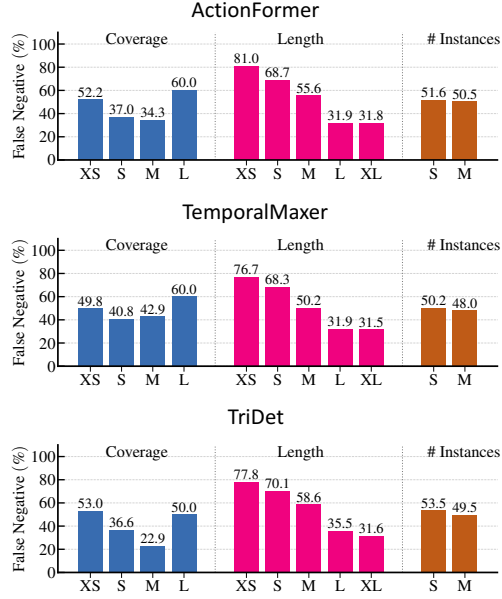


Figure B.16.: DETAD False Negative (FN) analysis of the TAL architectures [204, 167, 151] on the Wetlab dataset [149]. Experiments are those described in the results chapter of the main paper with the random seed being fixed at 1. The figure shows FN rates for different types of video segments. One can see that all architectures struggle with predicting small segments correctly.

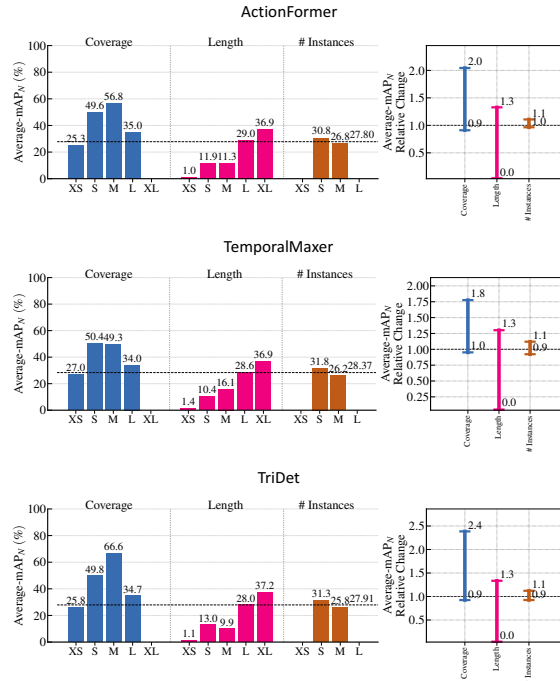


Figure B.17.: DETAD Sensitivity analysis of the TAL architectures [204, 167, 151] on the Wetlab dataset [149]. Experiments are those described in the results chapter of the main paper with the random seed being fixed at 1. Each subplot exists of a plot showing the mAP across different video segments (left) and as well as visualized change in mAP for the three segment categorizations (right). One can see that especially longer segments are predicted more reliably with especially extra small segments being not recognized at all.

B.5. Ablation Study on Hyperparameters of TAL Models

The following will provide a performed ablation study on the most significant hyperparameters, as specified by the original authors of the ActionFormer, TemporalMaxer and TriDet model. All experiments were conducted the same way as specified in the main paper, being the average across three runs using a set of three random seeds. To limit the amount of experiments we only assess results obtained on the WEAR dataset. All mentioned results are postprocessed applying a score threshold of 0.2.

B.5.1. Maximum Input Sequence Length during Training

The TAL models investigated in this paper all aim to predict all activity segments within an untrimmed sequence in one iteration, rather than batching the sequence like the inertial models do. In order to limit complexity, all three TAL models randomly truncate input sequence to not exceed a maximum sequence length during training. The maximum sequence length therefore becomes an important hyperparameter, especially when dealing with long initial input sequences per subject. Table B.4 shows results trying out different maximum sequence lengths for each of the three architectures being applied on the WEAR dataset. For reference, given a window size of 1 second with an overlap of 50%, per-subject sequences within the WEAR dataset contain on average more than 6000 windows per participant. Results show that in case of all architectures a small maximum input sequence length results in the highest classification and average mAP scores. Particularly, the TemporalMaxer benefits the most from a smaller input sequence length.

B.5.2. Feature Pyramid

One of the central parts of the ActionFormer, TemporalMaxer and TriDet model is the scaled feature pyramid. By applying a downsampling operator all three architectures aim to learn features at different temporal scales which can then be used to classify each timestamp based on a different temporal granularity. Table B.5 shows results when applying each model with different amounts of layers employed in the feature pyramid. Same as the ablation study performed by Shi et al. [151], we use both fixed and scaled number of bins when varying the number of layers in the TriDet model. Results show that the amount of layers significantly affect prediction performance across all architectures. In case of the TriDet and ActionFormer six layers, as also used in the main paper, are resulting in highest classification and average mAP scores. In case of the TriDet model, applying a scaled number of bins does result in better performance as more bins are needed to capture activities with a longer duration.

Table B.4.: Ablation results obtained by applying different maximum sequence lengths (T_{max}) for the ActionFormer (AF) [204], TemporalMaxer (TM) [167] and TriDet (TD) model [151] on the WEAR dataset [19]. Results show that a small maximum input sequence length results in the highest classification and average mAP scores. Note that the ActionFormer cannot be trained on a maximum sequence length of 1152, as the sequence length had to be divisible by the kernel sizes of the model. Best results per architecture are in **bold**.

	T_{max}	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
AF	512	70.67	76.88	71.36	0.22	7.09	0.64	8.27	0.01	1.70	71.71
	1280*	72.61	76.36	72.69	0.20	6.28	0.70	5.73	0.01	1.47	72.71
	2304	72.53	74.87	72.16	0.19	6.11	0.77	4.83	0.01	1.44	71.42
	4608	72.00	72.69	70.94	0.21	5.76	0.83	3.43	0.00	1.26	69.31
TM	512	69.34	73.67	69.57	0.22	7.43	0.77	7.04	0.02	1.22	69.45
	1152	71.02	74.10	70.88	0.19	6.48	0.81	5.28	0.01	1.12	71.23
	2304	69.29	70.79	68.51	0.20	6.38	0.93	3.70	0.01	1.01	67.59
	4608	66.05	65.78	64.40	0.22	5.83	1.09	4.00	0.01	0.97	61.48
TD	512	71.77	76.03	71.25	0.26	5.58	0.61	9.16	0.03	1.34	73.97
	1152	73.87	75.73	72.83	0.20	4.59	0.70	6.75	0.03	1.41	73.39
	2304	74.60	73.81	72.29	0.24	4.18	0.76	4.76	0.03	1.35	72.48
	4608	72.11	70.01	69.46	0.24	4.13	0.90	4.04	0.02	1.03	68.56

B.5.3. Number of Bins (TriDet)

As described in the previous section, the number of bins employed in the Trident head of the TriDet model influences the maximum length at which activities can be recognized. Table B.6 demonstrates that in case of the WEAR dataset an increase in bins improves both classification and average mAP scores, with the optimal value being 16.

B.5.4. Local Window Size for Self-Attention (ActionFormer)

The next set of experiments investigates the effect the size of the local self-attention window has on the results of the ActionFormer. As evident by Table B.7 results are only marginally affected by a changing window size. Though a larger window size of 25 does result in both higher classification scores and average mAP, improvements are less than a percent for both categories compared to the original window size of 9.

B.5.5. SGP Window Sizes (TriDet)

Similarly to the local self-attention window size, the TriDet model’s SGP layers are defined by two window size parameters, namely w and k . Table B.8 shows results when trying out different values for w and k , fixing one of them to 1 when trying out larger values for the other hyperparameter. One can see that the optimal value for w is between 9 and 13, while the optimal value for k is around 5. Nevertheless, as for the local self-attention, differences amongst results are only marginal with the initial setup ($w = 1$ and $k = 4$) being only

Table B.5.: Ablation results obtained by trying out different number of layers within the feature pyramid when training the TAL architectures [204, 167, 151] on the WEAR dataset [19]. Results show that the amount of layers significantly affect prediction performance across all architectures, with more layers needed to achieve good prediction results. In case of the TriDet model, applying a scaled number of bins does result in better performance as more bins are needed to capture activities with a longer duration. Best results per architecture are in **bold**.

	Levels	Bins	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
ActionFormer	1	-	23.90	17.61	17.91	0.34	4.96	2.66	3.68	0.03	0.08	6.78
	2	-	34.13	26.95	27.66	0.42	7.64	2.23	4.07	0.04	0.20	13.36
	3	-	44.90	41.95	41.32	0.38	10.86	1.76	2.83	0.02	0.61	28.91
	4	-	55.64	55.47	53.41	0.35	10.72	1.28	4.29	0.02	0.85	40.74
	5	-	66.55	69.89	66.34	0.28	9.31	0.86	5.01	0.01	0.88	59.20
	6	-	71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
	7	-	70.03	76.55	70.84	0.27	8.57	0.60	7.20	0.01	3.84	73.70
TemporalMaxer	1	-	27.81	20.41	20.84	0.43	6.27	2.44	6.55	0.04	0.10	7.93
	2	-	36.42	27.02	28.54	0.47	7.98	2.17	3.28	0.05	0.32	13.15
	3	-	42.01	38.25	37.73	0.39	10.30	1.87	4.03	0.03	0.74	24.20
	4	-	47.83	45.88	44.48	0.36	11.21	1.64	2.55	0.02	0.77	31.76
	5	-	58.48	57.84	56.24	0.32	9.29	1.27	2.92	0.03	0.63	46.70
	6	-	69.54	72.80	69.52	0.23	6.87	0.83	5.50	0.01	1.50	69.18
	7	-	68.52	75.53	69.77	0.24	8.14	0.68	5.61	0.01	4.37	71.78
TriDet	1	16	21.42	12.2	11.98	0.38	1.92	2.76	5.05	0.07	0.01	1.53
	2	16	34.53	22.24	24.07	0.49	4.57	2.34	3.32	0.05	0.08	9.08
	3	16	44.70	36.78	38.04	0.38	5.22	1.95	3.32	0.02	0.28	26.05
	4	16	59.86	52.80	54.01	0.31	4.51	1.42	3.14	0.06	0.19	42.45
	5	16	69.92	67.56	67.07	0.29	4.72	0.93	4.50	0.03	0.65	60.99
	6	16	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
	7	16	71.59	76.25	71.90	0.26	5.90	0.64	6.60	0.01	3.97	73.52
	1	512	22.16	20.18	19.13	0.26	5.80	2.65	5.15	0.02	0.53	11.47
	2	256	33.46	29.39	29.11	0.31	7.09	2.27	2.76	0.02	0.36	19.55
	3	128	44.90	38.69	39.31	0.33	5.85	1.92	3.61	0.03	0.24	29.35
	4	64	58.95	52.31	53.44	0.34	4.97	1.42	3.80	0.03	0.16	43.26
	5	32	69.71	67.29	66.89	0.29	4.97	0.93	4.63	0.03	0.67	60.42
	6	16	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
	7	8	72.49	76.92	72.50	0.27	5.65	0.60	5.91	0.01	4.31	73.80

Table B.6.: Ablation results obtained by trying out different number of bins when training the TriDet [151] on the WEAR dataset [19]. One can see that a larger bin count results in better prediction performance, with the optimal value being 16. Best results are in **bold**.

Bins	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
4	73.88	76.48	73.41	0.24	4.88	0.66	6.53	0.01	1.78	73.88
8	74.38	75.30	72.99	0.29	4.74	0.66	6.36	0.02	1.51	74.54
10	74.38	75.49	73.15	0.27	5.06	0.66	6.08	0.03	1.36	74.12
12	74.84	75.67	73.54	0.28	4.60	0.65	6.30	0.02	1.51	74.67
14	74.83	75.56	73.35	0.29	4.69	0.64	6.51	0.03	1.44	74.53
16	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
20	73.47	74.71	72.10	0.27	4.59	0.69	6.33	0.02	1.50	73.87

Table B.7.: Ablation results obtained by trying out different local self-attention window sizes when training the ActionFormer [204] on the WEAR dataset [19]. Though a larger window size of 25 results in both higher classification and average mAP scores, improvements are less than a percent for both compared to the original window size of 9. Best results are in **bold**.

Window Size	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
9	71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
19	71.38	76.20	71.84	0.22	6.64	0.67	7.45	0.01	1.90	73.84
25	72.13	76.81	72.50	0.22	6.51	0.66	6.78	0.01	2.03	74.07
37	71.50	76.32	71.86	0.24	6.87	0.66	6.96	0.00	1.89	73.63
Full	71.72	76.02	71.81	0.22	6.62	0.68	7.42	0.01	1.91	73.26

marginally worse than the optimized parameter values.

B.5.6. Loss Weight

As TAL architectures predict segments consisting of a class label and activity onsets and offsets, the models are optimized using both a classification and regression loss. Table B.9 explores different weighting schemes between the two losses. The weight λ_{reg} mentioned in the table refers to the relative weight of the regression loss in calculating the final loss of each of the three models. Results show that a larger regression loss weight results in more precise boundaries and higher mAP scores. Nevertheless, one can achieve the best trade-off between classification and mAP scores when using an unweighted, combined loss.

Table B.8.: Ablation results of trying out different hyperparameter values for both w and k in the SGP layers when training the TriDet [151] model on the WEAR dataset [19]. While optimal values for w and k are between 9 and 13 (w) and around 5 (k), while the optimal value for k is around 5, differences amongst results are only marginal. Best results per hyperparameter are in **bold**.

k	w	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
k = 1	1	73.91	74.92	72.49	0.30	4.74	0.65	7.32	0.03	1.53	74.54
	3	73.96	75.01	72.58	0.29	4.74	0.67	6.75	0.01	1.53	74.01
	5	73.83	75.01	72.67	0.28	4.50	0.67	7.30	0.02	1.48	74.04
	7	74.39	75.28	73.03	0.29	4.84	0.65	6.64	0.04	1.59	74.69
	9	74.99	75.68	73.51	0.29	4.87	0.65	6.63	0.02	1.47	74.90
	11	73.86	75.49	72.97	0.26	4.88	0.68	6.19	0.02	1.54	74.00
	13	75.73	76.35	74.40	0.26	4.60	0.65	5.69	0.02	1.50	74.43
	15	75.12	75.99	73.89	0.28	4.61	0.64	5.50	0.01	1.69	74.70
	17	73.96	75.08	72.88	0.30	4.97	0.66	6.68	0.01	1.55	73.89
	19	73.93	74.74	72.36	0.29	5.02	0.67	6.32	0.03	1.40	74.69
	21	74.33	75.23	72.67	0.30	4.88	0.64	6.21	0.03	1.76	74.68
w = 1	1	73.91	74.92	72.49	0.30	4.74	0.65	7.32	0.03	1.53	74.54
	2	75.31	75.64	73.61	0.29	4.52	0.63	6.66	0.02	1.62	74.67
	3	74.55	75.58	73.42	0.27	4.92	0.66	5.64	0.02	1.43	74.03
	4	73.84	74.80	72.36	0.29	4.78	0.67	6.63	0.03	1.45	74.29
	5	75.54	76.14	74.03	0.27	4.87	0.62	6.53	0.04	1.34	74.03
	6	74.27	75.38	73.02	0.28	4.96	0.66	6.46	0.02	1.48	74.54
	7	74.29	75.46	73.01	0.28	4.73	0.64	6.68	0.03	1.39	74.31
	8	74.63	75.87	73.50	0.26	4.68	0.67	5.89	0.02	1.69	74.72
	9	74.75	75.74	73.50	0.28	4.71	0.64	6.90	0.03	1.44	74.68
	10	74.75	75.28	73.00	0.28	4.80	0.66	6.04	0.02	1.36	74.32
	11	75.30	75.83	73.34	0.32	4.81	0.60	6.49	0.01	1.68	75.05

Table B.9.: Ablation results on trying out different relative regression loss weights λ_{reg} when training the three TAL architectures [204, 167, 151] model on the WEAR dataset [19]. Once can see that a higher regression loss weight results in more precise boundaries, though the best tradeoff amongst classification and mAP scores can be achieved when using an unweighted combination of regression and classification loss. Best results per architecture are in **bold**.

	λ_{reg}	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	Avg. mAP (↑)
ActionFormer	0.5	71.40	76.61	72.21	0.21	6.86	0.66	7.21	0.01	1.95	73.64
	0.2	71.9	76.36	72.03	0.23	6.97	0.65	6.45	0.01	1.84	73.06
	1	71.88	76.70	72.43	0.22	6.48	0.65	7.12	0.01	2.06	73.80
	2	72.73	76.96	72.72	0.24	6.68	0.64	6.87	0.01	2.07	73.58
	5	71.47	76.86	71.97	0.22	6.49	0.66	7.96	0.01	1.96	74.19
T. Maxer	0.5	68.81	71.59	68.44	0.23	6.99	0.86	5.22	0.01	1.46	68.03
	0.2	68.67	71.57	68.23	0.25	7.60	0.84	5.96	0.01	1.46	68.11
	1	69.54	72.80	69.52	0.23	6.87	0.83	5.50	0.01	1.50	69.18
	2	69.44	73.55	69.80	0.22	6.50	0.80	6.35	0.01	1.63	70.82
	5	69.61	73.98	69.49	0.25	6.99	0.73	7.94	0.02	1.72	71.00
TriDet	0.5	71.46	74.87	70.92	0.29	7.14	0.65	6.44	0.02	1.73	72.74
	0.2	68.62	74.47	68.92	0.27	8.44	0.68	6.93	0.01	2.40	70.30
	1	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	77.12
	2	74.16	75.29	72.61	0.27	4.65	0.66	7.09	0.02	1.56	74.44
	5	74.32	75.82	72.93	0.27	4.98	0.64	7.15	0.03	1.46	74.82

C. Supplementary Material: Weak-Annotation of HAR Datasets using Vision Foundation Models

C.1. Annotation Pipeline

The following provides figures and tables supplementary to the annotation pipeline experiments of the main paper.

C.1.1. Number of Samples per Cluster Clip Length

As detailed in the experiments section of the main paper, we used only the centroid clip to propagate labels to all other instances within the cluster. Table C.1 provides supplementary results comparing our approach with using not only the centroid, but also the labels of four other neighboring clips to determine the label of the cluster. In this approach, the four other instances are the closest neighbors to their respective centroid. We determine the label by majority vote, meaning the label of the centroid can be overridden if the majority of the other four clips is labeled otherwise. One can see that throughout all datasets, our approach remains stable compared to using more instances, suggesting that our label propagation based on the centroid clip is stable and that the centroid clip is indeed representative of other clips within the cluster.

Similarly, Table C.2 provides supplementary results when applying different clip lengths during clustering. As we mention in our main paper, we deem a clip length of four seconds to be adequate for an annotator to correctly label the activity present in the clip. While using longer clips would cause clips to potentially contain multiple activities, shorter clips would make it harder for a human annotator to correctly identify ongoing activities. Nevertheless, as also shown in Table C.2, a longer clip also equates to higher labeling accuracies across all datasets, suggesting activities can be more easily grouped together during clustering.

Table C.1.: Comparison of average labeling accuracy and standard deviation across study participants propagating labels based on only the human-annotated label of the centroid clip ($s = 1$) versus based on the 5 most-centered clips ($s = 5$). Each setting applied a GMM-based clustering using 100 clusters. One can see that both approaches yield the same results across all datasets [19, 149, 42].

	s	$c = 19$	$c = 50$	$c = 100$
		L. Acc	L. Acc	L. Acc
WEAR	1	50.02 (± 14.69)	75.28 (± 7.97)	83.96 (± 4.48)
	5	50.15 (± 14.67)	75.62 (± 7.94)	84.25 (± 4.34)
	s	$c = 9$	$c = 50$	$c = 100$
		L. Acc	L. Acc	L. Acc
Wetlab	1	27.23 (± 6.45)	54.30 (± 8.79)	66.23 (± 7.86)
	5	27.21 (± 6.47)	54.53 (± 8.94)	66.32 (± 8.19)
	s	$c = 20$	$c = 50$	$c = 100$
		L. Acc	L. Acc	L. Acc
ActionS.	1	36.13 (± 7.53)	50.50 (± 5.09)	57.29 (± 5.64)
	5	36.15 (± 7.46)	50.45 (± 5.26)	57.95 (± 5.78)

Table C.2.: Comparison of average labeling accuracy and standard deviation across study participants based on different clip lengths. One can see that 4 second video clips result in the best labelling accuracy across (almost) all datasets [19, 149, 42]. The table further provides the relative amount of dataset an annotator would need to manually label.

	Clip Length	L. Acc	% Data
WEAR	1 sec.	75.15 (± 5.93)	2.66
	2 sec.	79.11 (± 4.85)	5.32
	4 sec.	83.96 (± 4.48)	10.65
	Clip Length	L. Acc	% Data
Wetlab	1 sec.	64.80 (± 7.90)	0.36
	2 sec.	67.15 (± 6.54)	3.95
	4 sec.	66.23 (± 7.86)	15.79
	Clip Length	L. Acc	% Data
ActionS.	1 sec.	49.66 (± 3.84)	0.66
	2 sec.	53.37 (± 6.74)	3.32
	4 sec.	57.29 (± 5.64)	13.27

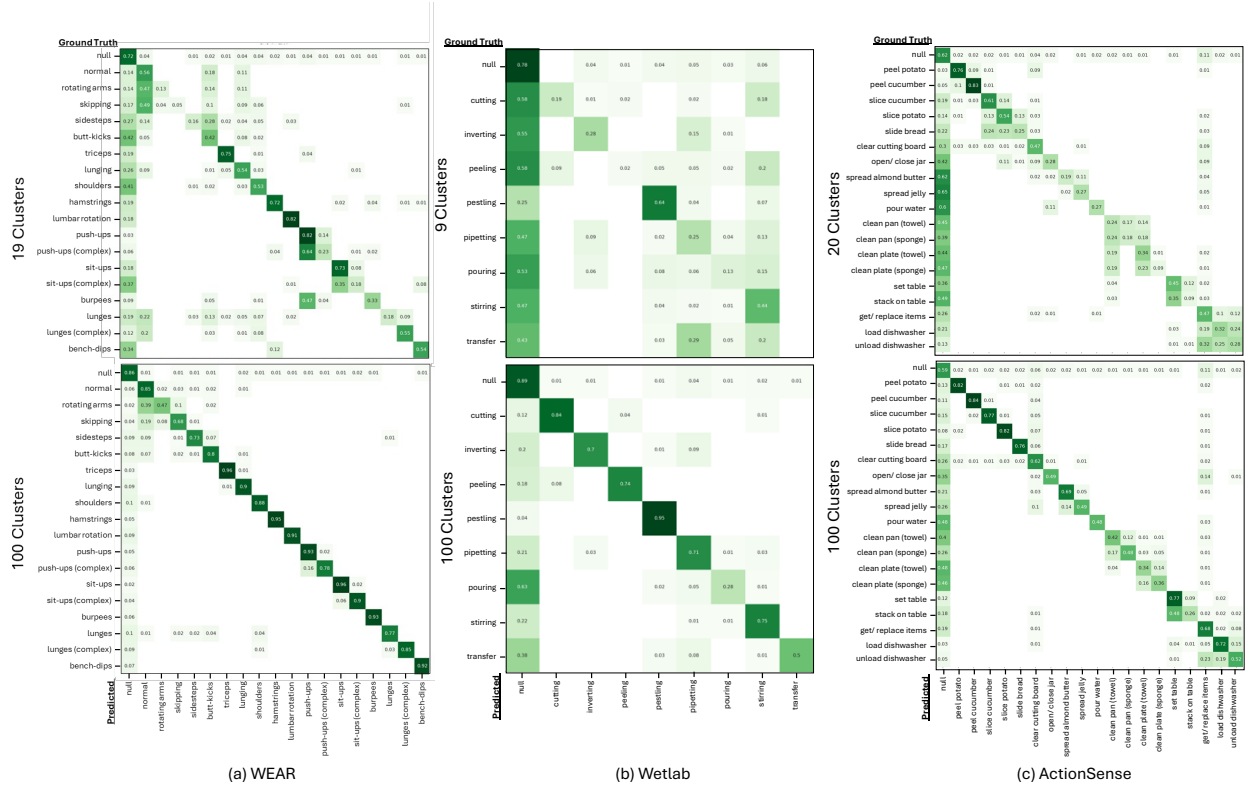


Figure C.1.1: Confusion matrices comparing the weak-labelling of the (a) WEAR [19], (b) Wetlab [149] and (c) Actionsense dataset [42] applying only (a) 19 (b) 9 (c) 20 versus (a-c) 100 clusters. One can see that with an increasing number of clusters, also similar activities can be differentiated.

C.1.2. Number of Cluster Per-Class Results

As mentioned in the main paper, we noticed that even though a dataset might not contain 100 activities, applying a higher number of clusters results in higher labeling accuracies. As Figure C.1 shows, when applying only a small number of clusters, similar activities tend to not be well differentiated (e.g., different styles of push-ups within the WEAR dataset).

C.1.3. Thresholding

As detailed in the main paper, we apply distance-based thresholding to eliminate outlier instances and increase the labeling accuracy within the cluster. Table C.3 details the results of applying three different thresholds (8, 6, and 4). One can see that the smaller the thresholded distance, the more data is being discarded. Nevertheless, we are capable of increasing the average labeling accuracy by a significant margin across all datasets. As shown in the weakly-supervised training section of the main paper, the thresholding significantly boosted the classifiers' performances (especially when applying only a small number of clusters).

Table C.3.: Average labeling accuracy and standard deviation across study participants applying different degrees of distance-based thresholding. Each setting applied a GMM-based clustering using 100 clusters. Though thresholding decreases amount of available training data, one can achieve a significant increase in labelling accuracy across all datasets [19, 149, 42].

	t	$c = 19$		$c = 50$		$c = 100$	
		L. Acc	% Data	L. Acc	% Data	L. Acc	% Data
WEAR	-	50.02 (± 14.69)	100	75.28 (± 7.97)	100	83.96 (± 4.48)	100
	8	53.80 (± 11.87)	64.89	76.57 (± 7.32)	86.89	84.24 (± 4.50)	95.25
	6	62.41 (± 9.02)	33.83	81.70 (± 4.71)	58.78	86.50 (± 4.57)	74.84
	4	87.15 (± 8.56)	11.19	91.07 (± 4.50)	22.51	93.01 (± 3.69)	33.76
	t	$c = 9$		$c = 50$		$c = 100$	
		L. Acc	% Data	L. Acc	% Data	L. Acc	% Data
Wetlab	-	27.23 (± 6.45)	100	54.30 (± 8.79)	100	66.23 (± 7.86)	100
	8	31.89 (± 10.09)	75.31	54.86 (± 9.00)	96.34	66.47 (± 8.04)	99.19
	6	47.94 (± 19.86)	37.97	62.31 (± 12.78)	72.62	70.03 (± 9.98)	85.79
	4	77.98 (± 20.72)	14.02	76.92 (± 15.15)	37.67	77.37 (± 9.83)	51.31
	t	$c = 20$		$c = 50$		$c = 100$	
		L. Acc	% Data	L. Acc	% Data	L. Acc	% Data
ActionSense	-	36.13 (± 7.53)	100	50.50 (± 5.09)	100	57.29 (± 5.64)	100
	8	40.20 (± 8.88)	56.27	53.37 (± 5.63)	80.87	58.51 (± 5.99)	85.14
	6	57.75 (± 10.58)	10.58	64.70 (± 8.05)	34.42	66.53 (± 6.16)	50.66
	4	77.40 (± 9.52)	3.73	80.42 (± 5.74)	8.23	79.84 (± 7.13)	7.13

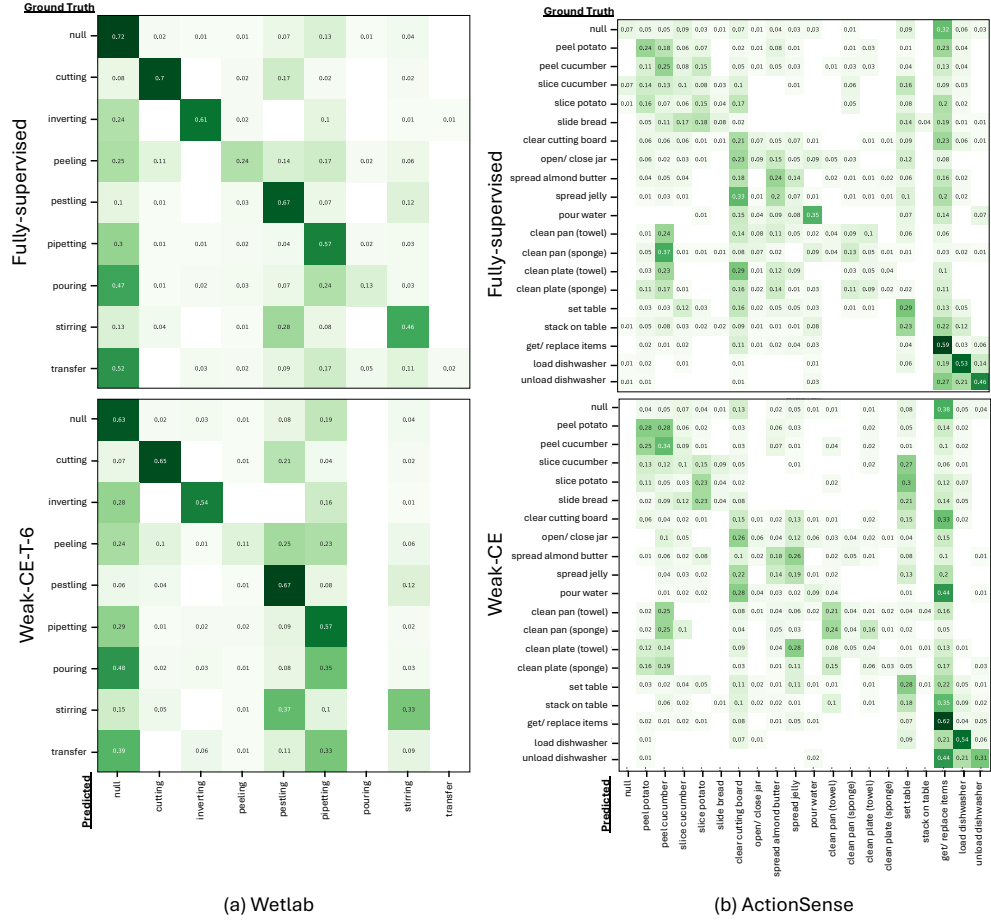


Figure C.2.: Confusion matrices comparing the shallow DeepConvLSTM [17] baseline results compared to that of the best performing weak-supervision approach on the (a) Wetlab [149] and (b) ActionSense [42] . With exception of the NULL-class, all activities were able to be classified close to the performance of the fully-supervised approach.

C.2. Weakly-Supervised Training

Figure C.2 provides a comparison of confusion matrices for the Wetlab and ActionSense datasets. As in the main paper, we compare the baseline results with the best-performing weak-supervision approach. One can see that, similar to the WEAR dataset, with the exception of the NULL class, all activities were classified close to the performance of the fully-supervised approach.